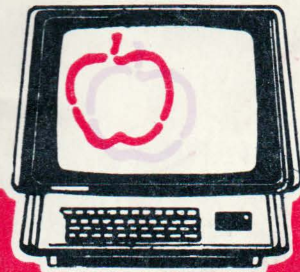
DJS-030系列微型计算机专刊



苹菓園

APPLE ORCHARD

5

1988

《苹果园》1988 第五期 总第二十七期

本 期 目 录

技术专文

DOS 3.3 系统环境下的数据压缩技术……姚政曙(1)

Apple PASCAL 磁盘文件管理与应用……解建民(5)

FORTRAN—80 语言级调用 6502 子程序……吴保芳(7)

一般位置园孔轴测图的 AppleII 机绘制……郭刚 罗继平(8)

AppleII 机用汇编语言给高级语言数组变量快速赋值方法……周凉(13)

数据如何在 VISICALC 与 BASIC 程序之间传送……李新(16)

在 Apple CP/M 操作系统中增加文件比较命令 COMP……施吉鸣(19)

实用程序

在 Apple soft 下超千万数值运算子程序……于建华 甘重(22)

大批量数据的快速排序……宋岳(24)

调试程序……郑平(25)

趣味程序

让 Apple 奏曲子……傅增明(29)

用 BASIC 语言解决 HANOI 塔问题……林峰(33)

辅助教学

＜抛体运动＞辅助教学软件……马基(35)

经验交流

87 年第 1 期《问题征答》参考答案……潘锐(25)

令 Apple II 显示目录命令改为 DIR……莫曰华(21)

一种更为简单的中文自启动程序……杨银洲(31)

Apple II 的表格打印……雷蕾(31)

中学生园地

用字符放大打印高分辨率图象的通用程序……刘善平(38)

用微机给知识竞赛计时计分……李书苓(39)

译文

介绍一种新型的微机 I/O 接口芯片……王应东(41)

软、硬件产品

CEC—I 中华学习机简介……王进吉(43)

Apple II 机、单板机通用数据转换(A/D)卡……李仲平(21)

苹果机 1.2 兆字节控制器通过设计定型鉴定……邱新之(52)

电脑应用

浅论 Apple 机在生油岩热解分析方面的应用……丛仁滋(44)

故障检测与维修连载

第九章 进一步的故障检测技术……(46)

编辑:《苹果园》编辑部

(山东省潍坊市东风大街 60 号)

出版:中国计算机用户协会 Apple 微机协会

主办:潍坊计算机公司

排版、印刷:潍坊计算机公司电脑排版中心

发行:潍坊计算机公司发行组

全年六期 欢迎订阅

本期用华光计算机激光汉字编辑排版系统排版

山东省报刊特许证:(鲁新出刊)字第 012 号

DOS 3.3 系统环境下的数据压缩技术

汕头市计算中心　姚政曙

摘　要

本报告提出一种 APPLE Ⅱ DOS 3.3 系统环境下的高速高效率数据压缩技术，讨论了组合特征型数据压缩和复原的原理，并给出了实施流程。以本技术实现的数据压缩，其压缩比高达 50%，存取速率高于现行 DOS 系统。更具有普遍使用意义的是：依本方法编制的压缩/复原程序仅占 216 个字节的长度，并作为现行 DOS 3.3 系统的一个补钉程序直接由 DOS 管理，从而形成了一个功能面向用户的高层次使用环境。文内所附实例表明，本技术的加入增强了现行 DOS 对数据资料的处理能力，而且不会对用户程序提出任何超越现行 DOS 系统规定范畴内的编程要求。

引言

减少数据资料对计算机储存空间需求的最有效方法是采用数据压缩技术。由于数据文件的直接开销是储存介质，而压缩后的数据文件只需较少的空间，从而可在软盘上储存更多的数据资料。鉴于微型计算机的大范围多领域的应用，对数据压缩技术的需要就更为迫切，这是由于微型机的外储存空间太小。这个问题对 APPLE Ⅱ 系统来说显得更为尖锐，因为 APPLE Ⅱ 所配置的驱动器其磁头鉴别率不高，故而只具有单密度软盘的储存能力，软盘的最大储存空间为 140K 左右，这一缺陷极大地限制了利用 APPLE Ⅱ 来作大批量数据资料管理方面的应用工作。加之软盘上的数据会由于时效作用或不当的操作而被损坏，所以通常要求对数据软盘至少拷备一份备份盘片，这样，大量数据的累积将造成软盘数量的急剧增加，不但增加了费用开销，而且对用户的软盘管理工作加重了负担。

基于此，Luber 在 1980 年发表于 creative computing 的一篇文章中首先提出了 APPLE 系统的有关图象数据压缩的方法，其原理类似于替代法的消零法，将数据中的连续重复部分在外储存器中消去，这种方法同样适用于数据文件中，但其压缩率取决于数据的重复度。而更多的数据压缩技术诸如 Lundon 的算术编码法[1]，Huffman 代码法[3] 等等，尽管比之 Lubar 法有效，但压缩和复原处理都非常复杂和耗时，且上述诸种方法均无法形成直接支持用户软件的系统操作环境，因而，这些方法的使用完全依赖用户对方法本身的接受能力，导致这些方法的应用实际上受到了极大的限制。

本报告提出的数据压缩/复原技术以及由此产生的名为“DC”的模块程序，采用的是组合状态和换值传送方法。该项技术完全符合 DOS 系统读/写数据文件的特点，因而，属于 DOS 系统的一个功能模块直接由 DOS 管理，这就形成了无需用户干预的系统高层次使用环境；压缩/复原一个字节耗时仅 30μs，却可节省磁头读/写及 DOS 的 FMS 系统耗费的近 100μs 时间；若不考虑数据的奇/偶效应，其压缩比恒为 50%，在奇/偶效应的影响下，压缩比略低于 50%。

符号表

Ain＝（　）——资料记录，表明记录 Ain 由集中的几个元素顺序排列组成，其长度为几个 8bit 字节。

D＝{　}——表明元素 D 且只可以代表集中的任何一个元素。

a_i——组成十进制数据的全部元素。

X_i——为 4bit 的可表征任意一个 a_i 的 16 进制值

$\$M_i$——8bit 长度的对应 a_i 的全部可供打印的 ASCII 码元素。

$Ni——由 4bit 的 Xi 和 X_{i+1} 的组合构成的 8bit 元素，Xi 为高位，X_{i+1} 为低位。

$Nj⊃$Mi，$ M_{i+1}——表示 $Nj 包含了 $Mi 和 $ M_{i+1} 两者的特征。

∀ $_i$——对一切 i。

Ln(A)——对 A 作长度为 n 的特征映象变换。

T 型文件的结构和传送

APPLE Ⅱ 系统的数据资料大多以 T 型文件读/写，T 型文件是最不复杂的文件结构，它是由许多资料记录 Ain 所组成，每个 Ain 之间由“回车”字符所分隔。16 进制值 $00 通常代表 T 型文件的结束，因此，在 T 型文件中，“回车”(ASCII 码为 $8D) 和 $00 这两个值具

有特殊意义，并对 DOS 的 POSITION（定位）和 APPEND（追加）这两个命令起标志作用，故尔，它们不允许被成为T型文件资料的一部分。同时，T型文件的传送必须经由 BASIC 系统中的 GET、INPUT 和 PRINT 指令来使资料与用户程序相沟通，因为这个原因而使得 APPLE 公司的 DOS 3.3 系统采用了等值传送的方法来传送记录 Ain，导致它所支持下的T型文件中的资料只能包含可以“被打印”出来的 ASCII 码字母，这个限制使得 APPLE Ⅱ 系统处理T型文件时速度较慢且软盘的空间利用率最差，因为数据资料中的每一个10进位数必须以一个表示其 ASCII 码的 8 bit 长度字节来储存。

从数据结构来看，如果不考虑资料与用户程序沟通的这一限制，对一个10进位数完全可用一个 NIBBLE 字节（即 4 bit 长度的字节）来表示，因为 4 个 bit 可表示 $0～$F 共 16 种状态。为了满足T型文件对 $8D 和 $00 这两个值的需要，在 NIBBLE 字节中排除 $0 和 $D 两种状态记录，则 4 个 bit 可表示 14 种状态记录。对于任意一个10进制实型数，其全部可能出现的数字元素是

$$a_i=\{0,1,2,3,4,5,6,7,8,9,+,-,\cdot,E\}\ \forall i \qquad (1)$$

式中，E 代表底数 10。

(1)式中的全部元素个数正好是 14 个，因此，与允许一个 NIBBLE 字节所代表的

$$NIBBLE=\{1,2,3,4,5,6,7,8,9,A,B,C,E,F\} \qquad (2)$$

14 种状态正好组成 1 对 1 的关系，显然地，两个 NIBBLE 字节组成一个 8 bit 字节，因而，一个 8 bit 字节完全能描述两个 a_i 即两个10进位数。如此，一个占有几位数（包括＋，－，·，E 在内）的10进制数据只应占有

$$m=n/2 \qquad (3)$$

个 8 bit 字节。据此，如果采用这种结构记录数据而又能被系统所接受，将比现行T型文件压缩软盘空间达 50%。

由于一个10进制数据所占有的位数可能是偶位数，亦可能是奇位数，而在奇位数状态下，为确保资料记录的间隔标志“回车”符不参与数据的压缩组合，最末一个 NIBBLE 字节必须被浪费，因而，(3)式应写成：

$$m=\begin{cases} n/2 & n\text{为偶数} \\ (n+1)/2 & n\text{为奇数}\end{cases} \qquad (4)$$

这就是在数据压缩过程中会发生的奇/偶效应，该效应实际上使压缩比只能接近于 50%。

组合特征型的数据压缩和复原

组合特征的方法由 Carl Banes 提出并用于集成电路、微处理器总线和数字电路的维修目的[2]，它的本质是以一个二进制值来表征一种特征状态，多个单独特征的二进制和即构成了一个组合特征。这种方法的优点是组合和分离是快速可逆的。而这种方法的设计思想被引进于本报告所提出的数据压缩/复原技术中。令：一个10进制数的原始数据构成形式为：

$$A_{on}=(a_1,a_2,a_3,\cdots,a_n) \qquad (5)$$

其中，a_i 为(1)式。而现行 DOS 3.3 系统环境下，对(5)式数据的读/写表现为下式：

$$A_{1n}=(\$M1,\$M2,\$M3,\cdots,\$Mn) \qquad (6)$$

式中，元素 $Mi 可能代表的 ASCII 码字母为：

$$\$Mi=\{B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,AB,AD,AE,C5\}\ \forall i \qquad (7)$$

由于

$$\begin{cases} Ain=Ln(Aon) & \forall_n \\ \$Mi=L(ai) & \forall_i \end{cases} \qquad (8)$$

所以，(6)、(7)两式仅仅是(5)、(1)两式的特征映象变换而已，在空间长度上并不产生任何畸变。很清楚，Ain 的结构类型满足等值传送的规范，而恰恰是该规范导致了数据储存空间的浪费。

设对(6)式作组合特征型压缩，则(6)式写成：

$$A_{2m}=(\$N_1,\$N_2,\$N_3,\cdots\$N_m) \qquad (9)$$

式中，

$$\$N_j=X_jX_{j+1} \qquad \forall_j \qquad (10)$$

为双 NIBBLE 字节特征的组合。并令：

$$X_i=L(\$M_i)=\{1,2,3,4,5,6,7,8,9,A,B,C,E,F\}\quad \forall i \qquad (11)$$

同时，为了强制

$$\$N_j\supset\$M_i,\$M_{i+1} \qquad \forall_j、\forall_i \qquad (12)$$

式成立，则当(6)式中 n 为奇数时，必须补充一个 $\$M_{n+1}$，且满足：

$$\$M_{n+1}\equiv AE \quad (\text{即小数点的}ASCII\text{吗}) \qquad (13)$$

由此得到(9)式中之 m 为(4)式。

(11)式是(7)式的特征映象变换，但由于 X_j，X_{j+1} 分别为一个 NIBBLE 字节，其结果导致资料记录的特征映象变换为：

$$A_{2m}=L_m\ (A_{1n}) \qquad (14)$$

故在空间长度上发生了压缩比 k≈1/2 的变化，这正是本报告所希望达到的结果。

为实现：

$$\begin{cases} A_{2m}=L_m(A_{1n}) \\ A_{1n}=L_n(A_{2m}) \end{cases} \qquad (15)$$

的可逆变换，令：

$$Y_i = L(\$M_i)$$
$$= \{01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0E, 0F\} \quad \forall_i \quad (16)$$

式中，Y_i 为 8bit 字节，于是有：

$$\begin{cases} Y_i = L(\$M_i) \\ Y_{i+1} = L(\$M_{i+1}) \end{cases} \quad \forall_i \quad (17)$$

再令：

$$Y_i = 16 * Y_i \quad 16为10进制值，下同 \quad (18)$$

同时令：

$$\$N_j = Y_i + Y_{i+1} = X_j X_{j+1} \quad \forall_i, \forall_j \quad (19)$$

实现了 $A_{2m} = L_m(A_{1n})$ 的变换。

复原变换是压缩的逆变换，已知 \$Nj，令：

$$\begin{cases} Y_i = \$N_j/16并取整 \\ Y_{i+1} = \$N_j - 16 * Y_i \end{cases} \quad \forall_i, \forall_j \quad (20)$$

取其逆变换

$$\begin{cases} \$M_i = L^{-1}(Y_i) \\ \$M_{i+1} = L^{-1}(Y_{i+1}) \end{cases} \quad \forall_i \quad (21)$$

即实现了 $A_{1n} = Ln(A_{2m})$ 的复原变换。

对于上述公式的运算及变换，在计算机中的执行是极其简单的。(17)、(21) 两式的变换仅需准备两份变换映象表即可，以被变换量作映象表的索引指针值，系统作一次脚标寻址运算即可获得变换后的映象。(18) 式是将 Y_i 连续作 4 次左移运算的结果，而 (19) 式则是将 y_i 和 y_{i+1} 作一次“或”运算获得。(20) 式中的 y_i 是将 $\$N_j$ 右移 4 位得到，而 y_{i+1} 是以 #\$0F 对 $\$N_j$ 作一次“与”运算的结果。压缩/复原两个字节资料仅需几十微秒，但每两个字节只向软盘写或读一个字节的压缩码，少占用 32 微秒的磁头动作时间，仅此一项，压缩/复原所耗费的时间已可被节省的时间所抵消了。

换值传送和 DOS 管理

要使本报告所列示的组合特征型数据压缩/复原技术面向用户，唯一的途径是使其成为现行 DOS 3.3 系统中的一个新的功能模块，以便掩盖它的实现和表达的细节，只将其根本目的直接显现出来，这样，就能使用户程序只需向系统“表态”而无需考虑“如何实现”，这也正是本项数据压缩技术优于其它文献资料报告的压缩技术之处。

这里，有两个困难：一是压缩数据如何与用户程序互相传送的问题，因为两者不具有共同状态，并且在传送方面不具有同等的自由。二是如何使这项压缩的读/写功能不经由 DOS 命令解释器 (DCI) 的解释而又能由 DOS 的档案管理系统(FMS)所识别并执行。

本报告所采取的技术是换值传送和 DOS 渗透。压缩数据模块由 DOS 系统中的最高层结构 DCI/SM (SM 为与 DCI 层次并列的状态器) 的 SM 处进入，这样就绕过了 DCI，从而无需为 DOS 系统创造要求用户另行掌握的新的 DOS 命令，而且，由于 SM 直接接受用户程序经由 PRINT 指令发送来的资料信息并能呼叫 DOS 的第二层结构 FMS 中“写一个资料字节”(W) 模块将资料发送给文件缓冲区，因而，在此可连续截取两个发送来的规范化资料字节 $\$M_i$，$\$M_{i+1}$ 并将其压缩成一个非规范化字节 $\$N_j$，从而以换值实现了接收用户程序资料的沟通。数据的复原模块则直接经由 DOS 的 FMS 中“读文件的一个字节”(R) 模块处进入，由于该 R 模块已读取了文件中送来的一个字节并仍由其发送给用户程序的 INPUT 指令，所以，在此截取压缩字节 $\$N_j$ 并再将其复原成 $\$M_i$ 和 $\$M_{i+1}$ 这两个 INPUT 指令能接受的规范化字节，以换值实现了向用户程序发送异型文件资料。由于该两模块必然进入 FMS，所以，压缩后的资料对软盘的 I/O 过程自然会由 FMS 呼叫 DOS 中最末层模块 RWTS 而完成。

有两个理由需要在上述两个模块中设立一个供用户程序“表态”的软开关，该软开关控制 DOS 系统对 T 型文件的读写采用 A1 方式(规范化数据格式)还是 A_2 方式(压缩数据格式)：

1. 在本报告的压缩技术实现之前，用户的 T 型文件均是在现行 DOS 下建立的，是规范化数据格式，必须保证本技术的加入仍能读/写规范化的 T 型文件，以保持系统的连续性。

2. 本压缩技术仅对数据资料有效，对文字符资料并不具备组合特征型压缩条件，所以对文字符资料记录仍要求以规范化方式读/写。

软开关以 BASIC 的 POKE 指令实现，本技术规定：

$$\begin{cases} POKE \quad L, \quad 0 \quad 要A1方式 \\ POKE \quad L, \quad 1 \quad 要A2方式 \end{cases}$$

这意味着，用户程序应在读/写 T 型文件的资料记录时以此条指令向系统作个表态，此种表态方式具有极大的灵活性，因为它同样允许用户程序在同一个 T 型文件中读/写 A1 和 A2 这两种异型结构的资料记录。用户“表态”在 DOS 已具有不止一种数据处理功能的情况下是不可避免的。

全部 DOS 渗透和数据的压缩/复原流程示于图。流程清楚地表明，依据本技术设计的 DC (DATA COMPRESSION) 模块，每读/写两个字节只访问一次 DOS 的 R/W 功能，因此，其 I/O 速率远高于现行 DOS。DC 模块以 6502 指令编制，仅占 216 个字节，程序是全浮动的，合理的按排可使其置于现行 DOS 的空洞区域

中。

实　例

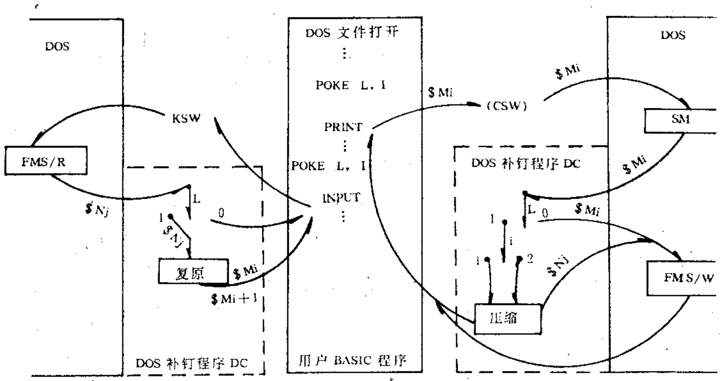
为了示范本数据压缩/复原的 DC 模块的使用并显示其效益，举实例如下(DC 模块已装入 DOS，用户表态软开关地址 L 为 10·进制值 969)：

编制用户程序如表 1。该程序的行 5～25 仅是为了产生两组数据相同的数组以被分别以 A1 方式和 A2 方式建立 T 型文件。行 30 设定以 A1 方式将数组 A 存在 T 型文件 A 中，行 60 设定以 A2 方式将数组 B 存在 T 型文件 B 中。行 90 和行 120 则分别设定以 A1 和 A2 方式读取文件 A 和 B 中的数据资料并分别存放于数组 A 和 B 中，行 150 以下为将读入的数组 A 和 B 中各自的前 4 个数据打印出来以比较两者是否一致。其余行内容均为现行 DOS 的读/写 T 型文件指令，不管是 A1 方式还是 A2 方式，两者没有任何差别。

运行该程序后打印出来的结果示于表 2，显然地两者的最终结果完全一致。

表 3 所示为用“CATALOG”命令列出的软盘目录清单，清单标明 T 型文件 A 占空间长度为 3 段而 B 则为 2 段。由于“CATALOG”是将任何文件都必须占有的至少为 1 段长度的 T/S 表也作为其空间累计的，所以文件 A 和 B 的资料实占段落数分别为 2 段和 1 段，文件 B 储存空间为文件 A 的 50%。

为了比较文件 A 和 B 在软盘上储存格式的差别，利用“读一个磁盘段落”的方式将两个文件的软盘储存信息直接取出，并分别列出其首端的几个字节资料示于表 4 中，由表 4 可见，文件 A 的数据结构为 A_{1n}，符合(6)式，而文件 B 的数据结构则为 A_{2m}，符合(9)式，这正是文件 B 所占空间大大少于文件 A 的根本原因。



L—用户表态软开关
i—压缩模块专用软开关，由系统控制
KSW/CSW—DOS 的 I/O 寄存器对

图　组合特征型数据压缩/复原流程图

该例表明：由于该项技术是面向用户的高层次操作环境，所以，用户程序完全不必考虑该技术的实现细节，实例程序清单完全证明了这一点。

表 1.　实例程序清单

```
5     DIM A(30): DIM B(30)
10    D$ = CHR$ (4):A(0) = 1
15    FOR I = 1 TO 30
20    J = I + 0.3:A(I) = -A(I-1) * J
25    B(I) = A(I): NEXT I
30    POKE 969,0
35    PRINT D$; "OPEN A"
40    PRINT D$; "WRITE A"
45    FOR I = 1 TO 30
50    PRINT A(I): NEXT I
55    PRINT D$; "CLOSE A"
60    POKE 969,1
65    PRINT D$; "OPEN B"
70    PRINT D$; "WRITE B"
75    FOR I = 1 TO 30
80    PRINT B(I): NEXT I
85    PRINT D$; "CLOSE B"
90    POKE 969,0
95    PRINT D$; "OPEN A"
100   PRINT D$; "READ A"
105   FOR I = 1 TO 30
110   INPUT A(I): NEXT I
115   PRINT D$; "CLOSE A"
120   POKE 969,1
125   PRINT D$; "OPEN B"
130   PRINT D$; "READ B"
135   FOR I = 1 TO 30
140   INPUT B(I): NEXT I
145   PRINT D$; "CLOSE B"
150       FOR I = 1 TO 4
155       PRINT "A=";A(I): PRINT "B=";B(I)
160       NEXT I
```

表 2.　程序运行结果

A=−1.3
B=−1.3
A=2.99
B=2.99
A=−9.867
B=−9.867
A=42.4281
B=42.4281

表 3.　软盘目录清单

DISK	VOLUME	254
A	003	HELLO
B	002	DT
B	002	BC
T	003	A

T 002 B

表 4. 文件 A 和文件 B 的软盘储存信息（部分）

```
0800— (C2 F4) 8D 3F AA 8D CA F9      B 文件
0808— 78 8D 53 F5 39 2F 8D C3
0810— 35 F9 79 A4 8D 25 27 F7
0818— 85 37 8D C2 14 52 F8 33
0820— 2F 8D 96 94 7F 3A 45 8D
```

注：括号中变为－1.3(下同)

```
0800— (AD B1 AE B3) 8D B2 AE B9      A 文件
0808— B9 8D AD B9 AE B8 B6 B7
0810— 8D B4 B2 AE B4 B2 B8 B1
0818— 8D AD B2 B2 B4 AE B8 B6
0820— B8 B9 B3 8D B1 B4 B1 B6
```

结 论

本数据压缩/复原技术采用组合特征方式，因而实现了快速逆变换，其压缩比高达 50%，实际读/写速率高于现行 DOS 的 T 型文件。由于采用了换值传送，所以成为现行 DOS 系统的一个补钉程序直接使 DOS 本身具备了这一压缩功能，形成了面向用户的使用环境，因而不对用户的实际使用带来任何困惑或麻烦。这项技术在 APPLE Ⅱ 系统上的使用将减少大量的软盘费用开销和机时费用开销，有普遍的使用价值和较高的效益。

这项数据压缩的技术原理同样适用于其它类型的微机系统。

参考资料

1. Clen G. Landon, An Introduction to Arithemtic Codin, IBM Journal of Research and Development, 1984. 28 (2), pp135—148
2. Carl Barnes, Composite Signature Save Time, Electronics Test, 1984. 4, pp24—32.
3. H · L · High land 著，中国计算机服务公司译，微型计算机系统安全技术，中国计算机服务公司，1985.
4. 姚政曙，组合特征型数据文件压缩/复原功能模块程序 DC，广东省汕头市计算中心技术档案，1985. 3.

APPLE PASCAL 磁盘文件管理与应用

陕西工学院 解建民

APPLE PASCAL 是 UCSD－PASCAL 系统在 APPLE 上的实现。本文介绍 APPLE PASCAL 磁盘文件管理的特点、内部结构和误删文件的恢复方法

一．APPLE PASCAL 的磁盘文件管理

APPLE PASCAL 的运行以软盘为基础，有一台软盘驱动器即可运行，最多可配置 6 台。APPLE PASCAL 使用 5 英寸单面单密度软盘，每片盘有 35 个磁道，每道 16 扇区，每扇区 256 字节。

APPLE PASCAL 把整个软盘分成 280 个逻辑块，编号为 0－279。每个逻辑块包含两个物理扇区，即每块 512 字节。第 0 块和第 1 块存放系统引导程序，第 2 块至第 5 块存放文件目录，第 6 块至第 279 块为程序或数据区。

文件目录区由 4 个逻辑块组成，每个文件在目录区占有一个目录项，每个目录项为 26 个字节。目录区的第 0 个目录项总是存放卷名(即每片盘的名称)和有关卷的信息。从第 1 个目录项开始存放文件目录，每片盘最多可存放 77 个文件。

卷目和文件目录结构如下表。

表 1 卷目录结构

相对字节	内 容
0－1	引导程序起始块号(为 0)
2－3	目录区结束块号＋1(为 6)
4－5	未用
6	卷名长度
7－13	卷名(最多 7 个字符)
14－15	盘逻辑块总数(为 280)
16－17	现存文件个数
18-19	未用
20－21	日期
22－25	未用

文件类型用第 4 字节的低三位形成 0－7 分别表示 BACK、BAD、CODE、TEXT、INFO、DATA、GRAF、FOTO 八种不现类型的文件。

日期由两个字节形成十六个二进制位，高七位表示两位十进制的年份，中间五位表示日数，低四位表示月份。

表 2 文件目录结构

相对字节	内　容
0—1	文件起始块号
2—3	文件结束块号+1
4—5	文件类型
6	文件名长度(文件删除时为 0)
7—21	文件名(最多 15 个字符)
22—23	逻辑长度
24—25	日期

APPLE PASCAL 进行文件操作时，以目录提供的信息作为依据。读文件时，依次在各目录项中查找要读文件的名称，查到时以此目录项给出的起始块号和结束块号+1 来控制读出文件。查目录项时，以卷目录给出的现存文件个数来控制最多查多少个目录项。换句话说，设卷目录中给出的现存文件个数为 X，则目录区中前 X+1 个目录项（第 0 个目录项是卷目录）反映了磁盘中现存文件的情况（以下称这 X+1 个目录项为有效目录项），从第 X+1 个目录项起其后的目录项是无意义的。每次删除文件时，APPLE PASCAL 修改目录区，包括修改卷目录中的现存文件个数和将原有效目录区中被删文件目录项后面的目录项前移，但并不改变盘中文件的内容。这样，文件目录项在目录区出现的先后次序和文件在盘上存放的次序总是一致的；盘上现存的 X 个文件总是连续地占用目录区的第 1 个至第 X 个目录项，而被删除文件的目录项并不保留在有效目录区中。

APPLE PASCAL 的磁盘空间分配策略是连续分配。建立文件时，总是寻找最大的自由磁盘空间，将其写入该自由空间的开头连续 n 个块中（n 视文件长度而定）。当该空间写不下文件时，给出“写不下”的信息而不建立该文件的目录项，但已写入盘中的该文件的内容并不作改变。写入成功时，建立目录项，文件便完整地留在磁盘中。

自由磁盘空间可以从目录表中反映出来，后一个目录项的第 0、1 字节表示的起始块号与前一个目录项的第 2、3 字节表示的结束块号+1 之差，即为这两个文件之间的自由空间块数。这样，各个目录项给出的起止块号已反映了整个磁盘空间的使用情况，APPLE PASCAL 仅依据文件目录就可进行磁盘空间的管理。

二. 磁盘文件的存放格式

不同类型的文件在磁盘中的存放格式是不同的。CODE 和 TEXT 是两种主要的类型。

类型为 CODE 的文件是由编译、汇编或链接程序产生的 P 码或机器码文件。CODE 文件的第 0 块存放段目录，其后的块存放代码文件内容。段目录结构见表 3。

表 3 段目录结构

相对字节	字节数/段	内　容
0—63	4	段存放的相对块号和段长度
64—191	8	段名
192—223	2	段类型
224—255	2	段接口文本存放的相对块号
256—287	2	版本号、代码类型、段号
288—291		该文件使用库单元情况
292—511		版权信息

段目录的主要特征是其中的段名（由字母、数字组成）可整齐地显示出来（通常用户程序为一段，其程序名出现在此）。CODE 文件的长度可从 0—63 字节中得到：找出最大的相对块号 B，将其后的段长度字节数除以 512 得商 C，有余数时将 C 加 1，则 B+C 为该文件长度。文件长度加段目录所在块号即得段目录中的结束块号。确定文件长度时应注意两点：①. 算得的结束块号不得大于文件所在自由空间的最末块号；②. 单独编译的单元和经汇编后的 CODE 文件长度除按上述计算外，应再加 1。

执行库文件建立程序 LIBRARY. CODE 产生的库文件，虽然用显示目录命令给出的类型为 DATA，实质上是 CODE 文件，其存放格式与 CODE 文件是一致的，SYSTEM. LIBRARY 便是一例。

类型为 TEXT 的文件由编辑操作或用户程序的 REWRITE (F，＜文件名＞) 产生（其中 F 说明为 TEXT）。TEXT 文件以每两块为一页（1024 字节）来组织。首页为标题页，存放编辑程序要用到的内部信息，其余页存放文件正文。TEXT 文件主要由 ASCII 码的可显示字符组成。

对于 TEXT 文件，只要按逻辑块号将盘上内容读出并显示，是不难找到文件在盘中存放的起始和结束块号的。对于 CODE 文件，只能从段目录中提供的信息来确定。

三. 误删文件的恢复

APPLE PASCAL 仅依据文件目录来管理磁盘文件。只要将误删文件的目录项放入目录区的适当位置，误删文件就可恢复。问题是经删除操作后，目录区被删文件目录项后的各目录项均已向前移动，被删文件的目录项已被其后的第一个目录项占据，如何能得到被删文件的目录项信息？其中包括文件名、文件类型、文件在盘中的起始块号和结束块号。对于误删文

（下转 12 页）

FORTRAN—80 语言级调用 6502 子程序

空军雷达学院　吴保芳

在 APPLE-Ⅱ 接口槽插入一块 Z80 卡。该机便可工作在 6502 又可工作在 Z80 的控制之下，而成为一个双 CPU 的微型计算机。可工作于 CP/M 操作系统。在 CP/M 系统下，正确调用 6502 子程序，对于充分挖掘微机潜力、优化和缩短应用程序显然是一种很有意义的事情。本文主要介绍在 FORTRAN-80 语言级如何调用 6502 子程序。特别是用 APPLE SOFT BASIC 的绘图子程序阐述我们的作法。

APPLE SOFT BASIC 具有绘图语句，这给使用者带来极大方便。BASIC 之所以有如此之功能，是因为主机上有固化了的有关绘图的解释程序。是否可以设想让 FORTRAN-80 也调用这些程序而同样具有绘图功能呢？回答是肯定的。不过 FORTRAN-80 不能直接调用 6502 机器语言子程序。需要另外一段汇编语言子程序作为 FORTRAN-80 主程序调用 6502 子程序的桥梁，再将该汇编子程序与 FORTRAN-80 主程序实施链接装配生成一个绝对文件以后，你的 FORTRAN-80 主程序便具有绘图功能了。这也算得上对 FORTRAN-80 功能的一个小小的扩展。以下详细说明，先列出几个 6502 子程序的首地址。

0F457H—画点子程序首地址

0F53AH—画线子程序首地址

0F399H—定义屏幕为文本方式的子程序首地址

0F3E2H—定义屏幕为 HGR 的首地址

0F3D8H—定义屏幕为 HGR2 的首地址

因为是 6502 的子程序，所以地址无需转换为 6502 内存的实际地址。

信息传送区的填法：

画点时，将竖座标置入 A 寄存器传送区。横座标低 8 位填入 X 寄存器传送区，高 8 位填入 Y 寄存器传送区（用 Z80 的 8 位寄存器对可自动完成这一工作），调用 0F457H，在屏幕上可得到一点。

画线时，先画起始点，横竖座标的填法如上，调用 0F457H；再将终点横座标的低 8 位填入 A 寄存器传送区，高 8 位填入 X 寄存器传送区。竖座标填入 Y 寄存器传送区。调用 0F53AH，这样就完成了由起点到终点画一条线。

下面是在高分辨率第二页绘制正弦波曲线的例子。

```
A>TYPE FORTRAN. FOR
          PROGRAM MAPE
          COMMON/D/J,K
          CALL SUBR1
          PAI=3.14159/90.0
          DO 100 I=1,180
          S=FLOAT(I-1)
          DEG=PAI*S
          CYC=SIN(DEG)
          J=INT(80.0-50.0*CYC+.5)
          K=24+I
          CALL SUBR2
100    CONTINUE
       PAUSE'WAIT'
       CALL SUBR3
       WRITE(1,50)
50     FORMAT('+GOOD! ')
       STOP
       END
```

```
                ENTRY SUBR1
Z$CPU           EQU 0F3DEH
A$VEC           EQU 0F3D0H
HGR2            EQU 0F3D8H
POINT           EQU 0F457H
LINE            EQU 0F53AH
TEXT            EQU 0F399H
A$ACC           EQU 0F045H
A$XREG          EQU 0F046H
A$YREG          EQU 0F047H
MEMORY:         COMMON/D/
SUBR1:          LD HL,HGR2
                LD (A$VEC),HL
                LD HL,(Z$PU)
                LD (HL),A
                LD IX,18H
                LD A,0
                LD (A$ACC),A
                LD DE,POINT
                LD (A$VEC),DE
                LD (HL),A
                LD (A$ACC),IX
                LD A,9FH
```

```
25:                  LD (A$YREG),A
26:                  LD BC,LINE
27:                  LD (A$VEC),BC
28:                  LD (HL),A
29:                  LD (A$XREG),IX
30:                  LD A,50H
31:                  LD (A$ACC),A
32:                  LD (A$VEC),DE
33:                  LD (HL),A
34:                  LD (A$YREG),A
35:                  LD IX,117H
36:                  LD (A$ACC),IX
37:                  LD (A$VEC),BC
38:                  LD (HL),A
39:                  LD (A$VEC),DE
40:                  RET
41:                  ENTRY SUBR2
42:     SUBR2:       LD A,(MEMDRY)
43:                  LD (A$ACC),A
44:                  LD BC,(MEMORY+2)
45:                  LD (A$XREG),BC
46:                  LD HL,(Z$CPU)
47:                  LD (HL),A
48:                  RET
49:                  ENTRY SUBR3
50:     SUBR3:       LD HL,TEXT
51:                  LD (A$VEC),HL
52:                  LD HL,(Z$CPU)
53:                  LD (HL),A
54:                  RET
55:                  END
```

主程序中 J=INT(80.0−50＊CYC+・5)，是为了在屏幕上得到适当比例的图形进行的变换，不影响曲线描绘的精度。K 为横座标参数。

PAUSE 可以让画面在屏幕上停留任意时间，按"RETURN"键则继续运行。

SUBRI　定义屏幕，画座标轴

12～15　定义屏幕为 HGR2

16～28　画竖座标轴。起点(24，0)，终点(24，159)

29～38　画横座标轴。起点(24，80)，终点(279，80)

39　存画点子程序地址，为了 $SUBR_2$ 画正弦曲线作准备。

$SUBR_2$　画正弦曲线，由 COMMON/D/传递座标参数。

SUBR3　定义屏幕为文本方式

几点说明：

1. 本汇编子程序可存在磁盘上，随时供 FORTRAN-80 链接使用。可根据波形的极性、比例仅需修改座标轴的起点和终点。

2. 若仅在 Z80 汇编语言级绘图而不与 FORTRAN-80 主程序链接，图形绘完后将因由 6502 返回 CP/M 而随即消失，可用如下指令使画面作任意时间的停留。

```
LD C,1H
CALL 5H
```

按任意键后程序继续运行。

3. 图形的硬考贝

将
```
LD HL,0FF69H
LD (0F3D0H),HL
LD (0E400H),A  ;Z80 卡插于 4 号槽
RET
END
```

作为一个文件经汇编链接存于系统盘上，运行后，系统进入 6502 监控"＊"，"CTRL-B"↙返回 BASIC"]"。然后按通常的图形拷贝方法进行。

4. 同 SOFTBASIC 一样，若程序长了，也会破坏高分辨率图形缓冲区，这里提请读者注意。

综上所述，CP/M 系统提供了调用 6502 子程序的功能，而且调用也比较简便。这样当 APPLE-II 工作于 CP/M 系统时，可免于一部分程序的编写。

一般位置圆孔轴测图的 APPLE-II 机绘制

吕潍农校　郭刚　自贡盐校　罗继平

引言

用计算机绘制圆孔的轴测图，难点在于隐藏弧的消隐。有的用联解椭圆方程的方法求出不完全椭圆可见弧的起始角和终止角，绘出了特殊位置圆孔的正等测轴测图〔1〕，但未解决一般位置圆孔任意正轴测图的绘制问题。因此，本文对这一问题进行了探讨。文中采用两次旋转变换的方式建立起一般位置圆孔的空间坐标数学模型，并提出以可见投影椭圆作为窗

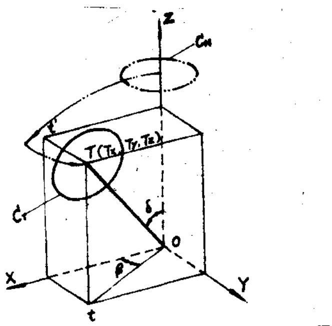
口对隐藏投影椭圆进行裁剪消隐，从而在 APPLE-II 机上实现了一般位置圆孔任意正轴测图的绘制。

程序允许圆孔轴线任意倾斜、投影方向任意角度的各种情况。绘图时，只需键入圆孔顶面圆圆心坐标和投影方向角，便可自动绘出圆孔的轴测图。

一、圆孔的空间坐标

为讨论方便，建立坐标如图一所示。设圆孔底面圆圆心位于坐标原点 O，顶面圆 C_T 的圆心位于一般点 T（T_x，T_y，T_z），轴线 OT 倾斜于投影面。OT 的水平投影与 X 轴的夹角为 β，OT 与 Z 轴的夹角为 δ，则

$$\left.\begin{aligned} \beta &= arctg(\frac{T_y}{T_x}) \\ \delta &= arctg(\frac{\sqrt{T_x^2+T_y^2}}{T_z}) \\ \text{孔深}|OT| &= \sqrt{T_x^2+T_y^2+T_z^2} \end{aligned}\right\} \quad (1)$$



图一

如果将一个与 CT 半径相同、深度相同的水平位置圆孔 CH 先绕 Y 轴正向旋转 δ 角，再绕 Z 轴正向旋转 β 角，则 CH 与 CT 完全重合（见图一）。因此，一般位置圆孔的空间坐标可由水平位置圆孔的空间坐标乘以两次旋转矩阵 Tr 而得到。下面采用四维齐次坐标进行变换。

$$[Tr]=\begin{bmatrix} cos\delta & 0 & -sin\delta & 0 \\ 0 & 1 & 0 & 0 \\ sin\delta & 0 & cos\delta & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\times\begin{bmatrix} cos\beta & sin\beta & 0 & 0 \\ -sin\beta & cos\beta & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$=\begin{bmatrix} cos\delta\cdot cos\beta & cos\delta\cdot sin\beta & -sin\delta & 0 \\ -sin\beta & cos\beta & 0 & 0 \\ sin\delta\cdot cos\beta & sin\delta\cdot sin\beta & cos\delta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

应注意到公式(2)是按点 T 在第一分角位置时写出的，当点 T 位于不同分角时，其旋转方向是不同的。根据〔2〕所分析的空间点位置与旋向的关系可知，第一次旋转时，若 T_x 与 T_z 同号，应正向旋转；若异号，为反向旋转。第二次旋转时，若 T_x 与 T_y 同号，为正向旋转；若异号，为反向旋转。所以，反向旋转时，应将公式(2)中的正弦函数项乘以系数 $L_i=-1(i=1,2)$。

设水平圆上点的坐标为（X_1,Y_1,Z_1），其中

$$\begin{cases} X_1=-R\cdot cos\alpha \\ Y_1=-R\cdot sin\alpha \\ Z_1=|OT| \end{cases}$$

则一般位置圆的空间坐标为

$$[X \quad Y \quad Z \quad 1]=[X_1 \quad Y_1 \quad Z_1 \quad 1]\times[Tr]$$

即

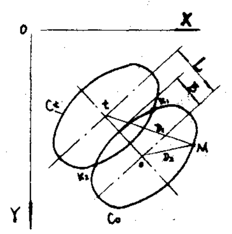
$$\begin{cases} X=X_1\cdot cos\delta\cdot cos\beta-L_2\cdot Y_1\cdot sin\beta+L_1\cdot Z_1\cdot sin\delta\cdot cos\beta \\ Y=L_2\cdot X_1\cdot cos\delta\cdot sin\beta+Y_1\cdot cos\beta+L_2\cdot L_1\cdot sin\delta\cdot sin\beta \\ Z=-L_1\cdot X_1\cdot sin\delta+Z_1\cdot cos\delta \end{cases} \quad (3)$$

在上式中，对于位于坐标面上的圆

$Z_1=0$

二、裁剪消隐原理

轴线倾斜于投影面的圆，其轴测投影一般仍为椭圆〔3〕。对于正圆孔而言，由于两个端面园位于一对相互平行的空间平面上，所以其轴测投影是一对长轴相互平行、轴线的投影重合于短轴的投影椭圆，见图二。



图二

设 C_t 为可见投影椭圆，C_o 为隐藏投影椭圆，L 为两椭圆中心距，B 为椭圆短半轴。可以看出，C_t 与 C_o 的关系有下列四种情况：

1° 相交 L＜2B

2° 相切 L＝2B

3° 分离 L＞2B

4° 重合 L＝0

其中，当且仅当 C_t 与 C_o 相交时，C_o 上才存在可见弧。

设 C_o 上任意点 M 到两个椭圆中心的距离分别是 $D_1=|Mt|$，$D_2=|Mo|$，显然

当 $D_1>D_2$ 时，M 为可见点

当 $D_1<D_2$ 时，M 为不可见点

当 $D_1=D_2$ 时，M 重合于交点 K_1 或 K_2，是可见弧

的端点。

因此，将可见投影椭圆（C_t）作为裁剪窗口，通过比较 D_1、D_2 的大小，即可判断出隐藏投影椭圆上诸点的可见性。仅将进入裁剪窗口的可见点画出，便可绘制出圆孔的消隐轴测图。

L，D_1，D_2 可根据两点距离公式

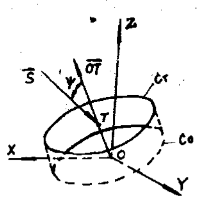
$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \qquad (4)$$

求出。

三、可见弧的存在条件

如（二）所述，仅当满足条件 L<2B 时，隐藏圆才具有可见弧。否则，属于全隐藏，不必进行 D_1、D_2 的计算与比较。

设 ψ 为承孔平面与轴测投影面之夹角（参见图三），则投影椭圆短半轴 $B=R\cdot cos\psi$



（图三）

因为承孔面法矢 $\vec{OT}=\{T_x,T_y,T_z\}$，轴测投影面法矢即投影线 $\vec{S}=\{S_x,S_y,S_z\}$，所以

$$\cos\psi=\frac{\vec{OT}\cdot\vec{S}}{|\vec{OT}|\cdot|\vec{S}|}$$

$\vec{S}$ 由投影角 $\varepsilon_1,\varepsilon_2$ 确定，令 $S_x=1$，$S_y=\text{tg}\varepsilon_1$，$S_z=\text{tg}\varepsilon_2$，则

$$\cos\psi=\frac{T_x+T_y\cdot\text{tg}\varepsilon_1+T_z\cdot\text{tg}\varepsilon_2}{\sqrt{T_x^2+T_y^2+T_z^2}\cdot\sqrt{1+\text{tg}^2\varepsilon_1+\text{tg}^2\varepsilon_2}} \qquad (5)$$

令 $CP=\cos\psi$，则得到隐藏圆可见弧的存在条件

$$L<|2\cdot R\cdot CP|$$

事实上，在绘图之前，应首先判断出 C_T 与 C_O 哪个是全可见圆周，以便将其投影椭圆作为裁剪窗口。由图三可以看出，当

$\psi<90°$（即 $cos\psi>0$）时，C_T 为可见圆

$\psi>90°$（即 $cos\psi<0$）时，C_O 为可见圆

$\psi=90°$（即 $cos\psi=0$）时，C_T、C_O 均积聚成直线。

四、编程粗框图

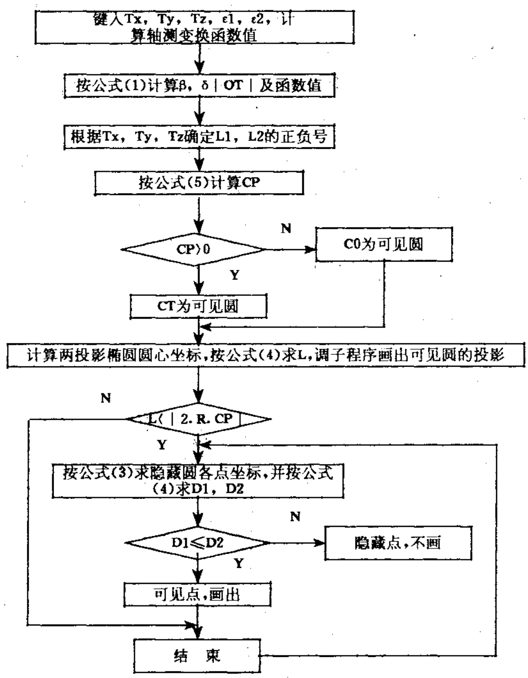


图　四

源程序：

```
5 PRINT CHR$ (4);"PR＃1"
10 REM DRAWING ALL HOLE
15 DIM IV(8,3),V(8,3),SV(8,3),S(6,5),N(6,3)
20 HGR : HCOLOR=3 : P=1.5708
30 HOME : VTAB 21: INPUT"TX,TY,TZ,S=";TX,TY,TZ,
   R
35 XA = 140:YA=80: IF TZ<0 THEN TX=-TX:TY=-
   TY:TZ=-TZ
40 INPUT "E1,E2=";E1,E2:F= 0:M = 0
50 GOSUB 900
60 T1=TAN(E1):T2=TAN(E2)
70 H=SQR (TX * TX+TY * TY+TZ * TZ)
100 IF TX =0 THEN BE=P:GOTO 120
110 BE = ATN (ABS (TY/TX))
120 IF TZ=0 THEN DE=P:GOTO 140
130 DE = ATN(SQR(TX * TX+TY * TY)/TZ)
140 SB = SIN(BE):CB=COS(BE):SD=SIN(DE):CD=COS
   (DE):L1=1:L2=1
150 IF SGN(TX * TY)<0 THEN L2=-1
160 IF SGN(TX * TZ)<0 THEN L1=-1
```

```
170 K1=SQR (TX * TX+TY * TY+TZ * TZ): K2=SQR (1+
    T1 * T1+T2 * T2)
180 CP=(TX+TY * T1+TZ * T2)/(K1 * K2)
190 IF CP< =0 THEN ZV=H:ZH=0:GOTO 210
200 ZV=0:ZH=H
210 X=TX:Y=TY:Z=TZ: GOSUB 430
220 IF CP< =0 THEN C1=XA:C2=YA:C3=XP:C4=YP:
    GOTO 240
230 C1=XP:C2=YP:C3=XA * C4=YA
240 Z1=ZH:G=1:GOSUB 300
245 LL=SQR ((C1-C3) * (C1-C3) + (C2-C4) * (C2-
    C4))
248 IF LL >=ABS (2 * R * CP) THEN 254
250 Z1=ZV:G=2:GOSUB 300
252 GOSUB 500
253 PRINT"PRINT THE PICTURE ? ":INPUT "P$=";P$: IF
    P$ >"P"THEN 255 RUN
254 PRINT CHR$ (4); "PR#1": POKE 1913,1: PRINT CHR$
    (17)
255 PRINT "A NEW PICTURE? ": INPUT "Y$="; Y$: IF
    Y$="Y" THEN 40
260 END
300 FOR A=0 TO 4 * P+.1 STEP.15
310 X1=-R * COS(A):Y1=-R * SIN(A):GOSUB 420
315 GOSUB 430
320 IF G=2 THEN 350
330 IF A=0 THEN HPLOT XP,YP:GOTO 410
340 HPLOT TO XP,YP:   GOTO 410
350 D1=SQR ((C1-XP) * (C1-XP) + (C2-YP) * (C2-
    YP))
360 D2= SQR ((C3-XP) * (C3-XP) + (C4-YP) * (C4-
    YP))
370 IF A=0 AND D1<=D2 THEN HPLOT XP, YP: GOTO
    410
380 IF D1>D2 THEN F=1:GOTO 410
390 IF D1<=D2 AND F=1 AND M=0 THEN HPLOT XP,
    YP:M=1: GOTO 410
400 HPLOT TO XP,YP
410 NEXT A: RETURN
420 X=X1 * CD * CB-L2 * Y1 * SG+L1 * Z1 * SD * CB: Y
    =L2 * X1 * CD * SB+Y1 * CB+L2 * L1 * Z1 * SD * SB:
    Z=-L1 * X1 * SD+Z1 * CD
425 RETURN
430 XP=XA-X * CG+Y * SG: YP=YA+X * SG * ST+Y *
    ST * CG-Z * CT: RETURN
500 REM SOLID
520 INPUT "L,W=";L,W
525 IF  (L / 2) < = R  OR  (L / 2)  (= R  THEN  PRINT
    "REINPUT":GOTO 520
530 FOR I=1 TO 8: FOR J=1 TO 3
535 ON J GOTO 540,560,580
540 IF I<5 THEN AA=L/2: GOTO 600
550 AA=-L/2: GOTO 600
560 IF I=1 OR I=4 OR I=5 OR I=8 THEN AA= -W/2:
    GOTO 600
570 AA=W/2: GOTO 600
580 IF I=1 OR I=2 OR I=7 OR I=8 THEN AA=0: GOTO
    600
590 AA=H
600 IV(I,J)=AA:NEXT J,I
610 FOR I=1 TO 8
620 X1=IV(I,1):Y1=IV(I,2):Z1=IV(I,3):GOSUB 420
630 V(I,1)=X:V(I,2)=Y:V(I,3)=Z
640 SV(I,1)=V(I,1) * CG-V(I,2) * SG
650 SV(I,2)=V(I,1) * SG * CT+V(I,2) * CG * CT+
    V(I,3) * ST:SV(I,3)=V(I,3) * CT-V(I,1) * SG
    * ST-V(I,2) * CG * ST
660 NEXT I
670 FOR I=1 TO 6: FOR J=1 TO 5
680 READ S(I,J):NEXT J,I
685 RESTORE
690 DATA 1,2,3,4,1,2,7,6,3,2,3,6,5,4,3,5,6,7,8,5,4,
    5,8,1,4,1,8,7,2,1
700 FOR I=1 TO 6:SA=S(I,1):SB=S(I,2):SC=S(I,3)
710 U1=SV(SB,1)-SV(SA,1):U2=SV(SB,2)-SV(SA,
    2):U3=SV(SB,3)-SV(SA,3)
720 V1=SV(SC,1)-SV(SA,1):V2=SV(SC,2)-SV(SA,
    2):V3=SV(SC,3)-SV(SA,3)
730 N(I,1)=U2 * V3-U3 * V2:N(I,2)=U3 * V1-U1 * V3:
    N(I,3)=U1 * V2-U2 * V1:NEXT I
740 FOR I=1 TO 6
750 IF N(I,2)<=0 THEN NEXT I
755 IF I>6 THEN 790
760 HPLOT XA-SV(S(I,1),1),YA-SV(S(I,1),3)
770 FOR S=1 TO 5: HPLOT TO XA-SV(S(I,S),1),YA-
    SV(S(I,S),3): NEXT S
780 NEXT I
790 RETURN
900 REM WORK OUT E1,E2,GA,TH
910 E1=E1/57.2:E2=E2/57.2
920 IF E1=0 AND (E2>1.57 AND E2<1.571) THEN GAM
    =0:GOTO 940
930 GAM=1.5708-E1
940 IF E2>1.57 AND E2<1.571 THEN THETA=1.5708:
    GOTO 960
950 THETA=ATN (TAN (E2) * COS(E1))
960 SG=SIN (GA):CG=COS (GA):ST=SIN (THETA):CT
    =COS (THETA)
```

970 RETURN

在源程序中，10—430 行是绘制一般位置圆孔轴测图功能段；900—970 行是子程序，计算轴测变换函数值功能段；500—790 行是绘制承孔板(六面体)轴测图功能段，其中

500—600 行：键入 L，W，计算水平承孔板各个顶点的坐标。

610—660 行：将水平板坐标转换成一般位置板的空间坐标，并进而转换成荧屏坐标。

670—790 行：计算各平面的外向法矢 $\vec{N}$，判断各表面的可见性，绘出可见面。

运行实例：

RUN↙

TX,TY,TZ,R=10,20,15,40↙

E1,E2=30,55↙

L,W=120,90↙

PRINT THE PICTURE ?

P$=P↙

A NEW PICTURE ?

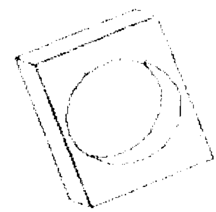
Y$=Y↙

E1,E2=50,120↙

可画出并打印一新的图形。

五、结语

本方法解决了一般位置圆孔任意正轴测图的自动绘制问题，程序具有一般性。对于轴线垂直于或平行于坐标面的特殊位置圆孔及其常用图形(正等测图、正二测图、三视图等)，只不过是本问题的特例而已，绘图时，只要输入相应的 T_X、T_Y、T_Z 以及投影角 E_1、E_2 即可得到。



图五

参考文献

〔1〕. 卢振荣：《计算机绘图初步》，西安交通大学出版社，1985

〔2〕. 曾大民：《实用计算机绘图》，华南工学院，1986

〔3〕. 山东工学院制图教研室：《轴测投影学》，山东科投出版社，1982

(上接 6 页)

件，通常知道其名称和类型，这两项不是主要问题。只要经删除操作后未对盘进行任何写操作，被删除文件一定在盘的自由空间中。对盘的自由空间依次扫描，根据各种类型文件的存放格式和自由空间各块的内容即可确定被删文件的起始和结束块号，然后按文件目录项的格式重组一个目录项，将其插入目录区中，并将卷目录中的现存文件个数+1，被删文件恢复了。重组的目录项插入目录区的位置应同时满足：①. 重组目录项中的起始块号应大于或等于其前一个目录项的结束块号；②. 重组目录项中的结束块号应小于或等于其后一个目录项的起始块号。

四．特定情况下文件的恢复

当误删文件目录项为原有效目录区的最后一个目录项时，删去该文件只是将卷目录中现存文件个数减 1，并将删去文件的文件名长度字节改为 0。这时，新有效目录区后的第一个目录项即为删去文件的目录项，要恢复此文件，只需按删除的逆过程操作。

当已知误删文件目录在目录区中的位置和文件的块数时，可用文件操作级的 M 命令直接恢复。方法是：用文件操作级的 E 命令得到详细的盘目录，确定误删文件落在哪一个<UNUSED>区内，若其余的<UNUSED>区显示的块数更大，则先用 M<假文件名>[块数] 造几个假目录，直到误删文件所在的<UNUSED>区最大为止，此时用 M<文件名>[块数] 即可将误删文件恢复，最后删去假文件。

结 束 语

APPLE PASCAL 提供了开发系统和应用程序的良好环境。实践证明，上述方法是可行的，但恢复误删文件成功与否取决于对 APPLE PASCAL 磁盘文件管理的了解，特别是对文件在磁盘中存放格式的掌握。

APPLE-II 机用汇编语言给高级语言数组变量快速赋值方法

四川大学计算中心　周　凉

摘　要：本文详细介绍了 APPLE—II 微机用汇编语言给高级语言数组变量快速赋值的两种方法，有一定的深度和技巧性，对从事微机应用工作的同志有一定参考价值。

一　引　言

随着计算机应用的发展，微型计算机越来越多地应用于工业控制和数据处理系统中，在这些应用中，如何节省内存，提高运行速度和编程效率，已成为不可忽视的问题。高级语言有较强的科学计算、图形显示和数据处理能力，用于分析计算很方便，较汇编语言编程效率高得多，但在高速数据采集和实时处理领域，则力不从心。另一方面，汇编语言运行速度快，占用内存少，但进行复杂的科学计算和数据处理比较困难，编程周期长，速度慢，修改维护困难，因此，很自然想到把汇编语言的快速功能与高级语言的科学计算，数据处理，图形显示功能相结合，以弥补各自的缺陷，使两种语言各自的优势得到发挥，以达到满意效果，下面介绍的两种用汇编语言给高级语言的数组变量快速赋值的方法，在计算机应用中经常遇到而又必须加以解决，因而它具有实用意义。

二、在汇编语言程序下快速给高级语言的整型数组变量赋值的方法

微型计算机用于实验数据的实时处理时，首先由采集器将模拟信号转换为数字信号，再将数字信号送入微机进行图形显示和分析处理。实际工作中，常在汇编语言程序下把给 A/D 转换后或直接从采集设备来的数据送入微机内存开辟的缓冲区（对快速数据采集往往这样的），缓冲起始地址的选择要避开 BASIC 解释程序区、用户程序区和汇编语言区。然后把数据赋给 BASIC 程序的数组变量，以便用高级语言进行处理，提高编程效率。一般采用 BASIC 语言中的 PEEK 函数来进行赋值，这种方法很方便，但若数据量很大，把这些众多的数据都采用 PEEK 函数给数组变量赋值，花费的时间相当可观。提高速度的方法是在汇编语言程序中查找数组变量的关键字，找出数组的起始地址，然后直接进行数据传输。APPLESOFT BASIC 语言的变量区安排在 LOMEM 以上和 HIMEM 以下的区域，从 LOMEM 指针开始由低向上存放简单变量、数组变量以及字符串组变量。对于一维的整型数组在各元素单元前占用 7 字节以说明数组变量（两个字节为数组名，两个字节用于数组大小，一个字节存放数组的维数，两个字节存维数的大小），然后是各元素单元，每一元素单元占用两个字节，高位在前，低位在后，依次类推，见图 1。数组变量名是以 ASCII 码存于内存，对于整型变量，每一变量名的 ASCII 码的最高位（第 7 位）置 1，例如整型。变量 A%的 ASCII 码为 C180。知道了数组表的结构方式及变量名的存放格式，就可用汇编语言直接对其变量赋值了。汇编语言赋值子程序框图如图 2。对于实型数组不能简单的使用这种方法，它的存放结构不同于整型数组，传送方法作相应改进后，也可使用此种方法，但非常复杂。不过 A/D 采集器所给出的数字量大都是整型量，因此能满足一般要求。

变量名	最高位为1	表头占7个字节
	最高位为1	
	数组总长低位	
	数组总长高位	
	维数	
	维数大小低位	
	维数大小高位	
	数，高位	一个元素占两个字节
	数，低位	
	数，高位	
	数，低位	
	…	

表 1. 一维整型数组表结构方式

图 2 只是对 8 位数据赋值的子程序，对于大于 8 位的数据，只稍作一点修改即可。某些 A/D 转换器给出的数字量为 BCD 码，将 BCD 码转换为 16 进制码就可采用此法。因此这种方法适合于一切用 A/D 转换器作输入设备，对其数据快速赋值给 BASIC 整型数组变量的情况。图 3 给出了主程序框图。

三、BASIC 语言与汇编语言结合给高级语言的实型数组变量赋值的方法

上面介绍的第一种方法，它适用于由 A/D 转换得来的数据，这些数据大多以 16 进制或 BCD 码表示，属于整型量，因此直接用 PEEK 函数赋值或用汇编语言快速赋值，都是很方便的。但是如果数据是从串行通讯接口板采集的，这些数据一般都用 ASCII 码表示，不同的仪器设备有不同的数据格式，而且是实数，有小数点，还有正负号，用第一种方法显然是不可能的，若经

过换码，换成 APPLE-II 的浮点数表示，再进行赋值，这也是相当复杂的，且效率限低，一般人很难实现。为此，本文提出了一个有效的解决办法，用于把串行接口 RS-232C 接收的数据交给 BASIC 语言处理，获得了很好的效果。

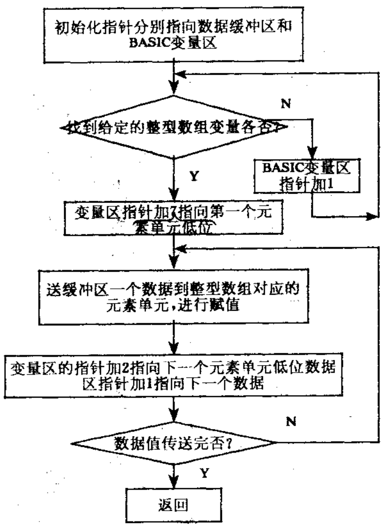


图 2. 汇编语言赋值子程序框图

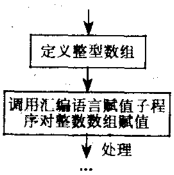


图 3　主程序框图

这种赋值，主要是要知道其 BASIC 程序缓冲区在何位置，怎样安排，数据区的数据格式等。APPLESOFT BASIC 程序在程序区的存放格式，是一种链式结构，换言之，找到一行程序，就可知道下一行程序在何地方。程序区是从 $800 开始的，程序区以一行程序作为基本单位存放的。这个基本单位开始的两个字节，存放一个链指针，指明下一程序行从内存中的那个单元开始（也就是下一个链指针的首地址在内存中的地址单元号码），基本单位的随后两个字节是本程序行的行号，接下去是程序行中的内容，除了 BASIC 保留字用内部代码外，其余码是各字符的 ASCII 码，一个程序结束时敲入的 RETURN，在此用 0 代表，任何一个程序行从键盘缓冲区转入程序区之后，程序区总存在一个末程序行，它是没有后续行号的。所以通常把它的链指针指向的地址及其后续地址（本应放下一个程序行的链指针的两个单元），填入 0 作为程序结束的标志，程序区的这种链式结构，示于图 4。

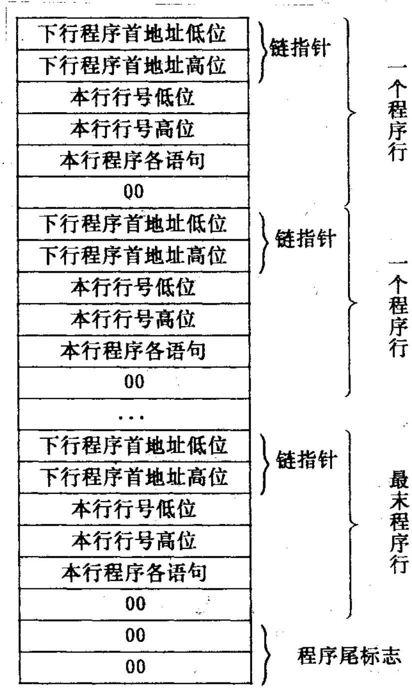


图 4. 程序区链式结构图

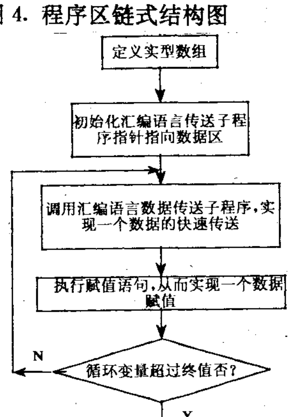


图 5　赋值程序框图

了解上述程序区结构后，还需了解数据区的格式（如每个数据的长度等），就可进行赋值操作。首先找到赋值语句赋值号“=”在程序区的位置，赋值号后预先要留出每个数据的长度。向赋值号后的地址单元传

送一个数据，执行赋值语句，即实现了一个数据的赋值，重复此过程，即可实现对整个数组赋值了。数据赋值程序框图见图5，汇编语言数据传送子程序框图见图6。

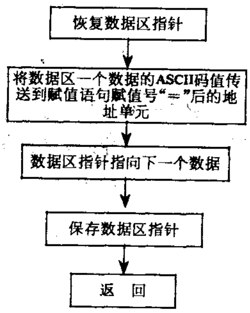


图6　汇编语言数据传送子程序框图

从图5赋值程序退出后，数据缓冲区的数据就赋给实型数组变量，以后便可以对数据进行所需处理。此种方法，作者在APPLE与MODEL340分光光度计联机进行数据处理中已成1实现，同时用APPLEII与MCC/340酶标比色计联机进行数据处理也采用了这种方法，它大大加快了编程速度，且很可靠，对从事这方面工作的同事 一定的启示。

四、结束语

近几年来，国内相继引进了许多带有EIARS－232C串行接口的多功能、高精度的智能仪器这些仪器和计算机相连，就能增加其功能、提高工作效率，为实验室自动化提供了可能，也使通用微机发挥更大的作用。因而本文提供的方法有一定的实用价值。

（上接18页）

```
2170  PRIT 1;",";0
2180  PRINT Q$;X$(I);Q$
2190  FOR J=1 TO V-1
2200  PRINT "V"
2220  NEXT J
2230  PRINT 1;",";0
2340  PRINT Q$;"  ";Q$
2250  V1=0:V2=MX(I):S$="V"
2260  GOSUB 2330
2270  NEXT I
2280  V1=-1 :V2=0:S$="EOD"
2290  GOSUB 2330
2300  PRINT D$;"CLOSE"
2310  RETURN
2320  PRINT T$
2330  PRINT V1;",";V2
2340  PRINT S$
2350  RETURN
```

这里，2030－2040句是开写顺序型数据文件命令；2050－2130句写前导记录；2140－2270句写数据记录；2280-2290句写结束记录。

我们将三、中的读RAIN、DIF程序加入：

```
240  GOSUB  1000
250  GOSUB  2000
```

以及上述两个子程序后，就组成一个传送并处理数据的完整程序。它运行后，将产生一个新的DIF文件，设名为RAIN1、DIF。容易用Visicalc的命令，将RAIN1、DIF读入Visicalc：

CITY-A	5.2	4.33	5.32	14.85
CITY-C	3.5	3.64	2.3	14.08
CITY-B	4.1	6.4	4.18	10.04

在这个新的城市降雨量的表中，增加了最后一列，是各城市三个月的平均降雨量；而且，城市也已按平均降雨量的大小顺序排列。

六、结束语

掌握DIF文件读、写方法后，应用上，我们可以编制读、写DIF文件的子程序，以及一些层次更深的计算或处理变量的子程序。适当组织它们为一专用程序，通对专用程序就能方便、有力地利用Visicalc，解决某一具体的科技、工程问题。

诚然，DIF文件不单为沟通BASIC语言而设计，它也是与其他高级语言，如FORTRAN、COBOL等，进行数据传送的接口。它们之间数据信息的交换方法是相同的。

数据如何在Visicalc与BASIC程序之间传送

福建水利电力学校　李　新

一、引　言

Visicalc等一类电子表格程序，是微型计算机最流行的应用软件之一。它最大的特点，是输入、输出数据，如同人们平常填写表格式地进行，并且各层次命令简单，随时反应于屏幕；十分直观、易学。Visicalc还具有相当强的计算能力，尤其当表格中某单元数据改变时，将自动地反映到与这单元数据有牵涉的其它单元的数据。所以，在商业、财务和办公室等方面，是一个很有力的工具。然而，对于科技、工程上的计算，往往产生大量中间数据，或者须对数据进行更强的逻辑处理，Visicalc就显得不是那样简便有力，甚至对一些特殊的数学函数的计算，更是无能为力。

但是，如果我们把Visicalc与高级语言结合使用，困难将得到解决，而且还可以充分发挥Visicalc的长处，这将大大加强Visicalc的功能。

Visicalc与高级语言结合使用，实际上就是解决它们之间数据传送的问题。本文讨论Visicalc、Magicalc这类电子表格程序与BASIC程序之间数据传送的方法。

二、数据信息交换的接口——DIF文件

Visicalc能够生成一种特定格式的顺序型数据文件，即DIF(Data Interchange Formate)文件，用来在内部或对外部进行数据信息的传送。所以，首先必须弄清DIF文件的结构。

DIF文件由三个部分组成：前导记录、数据记录和结束记录。

(1)前导记录(Header Record)

前导记录记载整个数据集合的形式结构及其参数，格式中下：

```
TABLE   }
0，1    } 第一组(前导记录开始的标志)
"  "    }

VECTORS }
0，V    } 第二组(数据集合向量，
"  "    } V为行向量或列向量的个数)

TUPLES  }
O，t    } 第三组(数据集合元串t为元串的个数)
"  "    }

DATA    }
O，O    } 第四组(前导记录结束标志)
"  "    }
```

以上可见，它包括四组，每组各三项。第一项是标题；第二项含两个由逗点隔开的数值；第三项是由引号括着的空字串。事实上，第一组与第四组的全部资料，只是分别用以标识前导记录的开始与结尾，而第二组与第三组则描述数据集合结构形式，即怎样排成矩阵表。

我们知道，电子表格中的资料，总是排成矩阵形式。例如，三个城市在第一季度三个月测得的降雨量输入Visicalc后，矩阵表为：

CITY-A	5.2	4.33	5.32
CITY-B	4.1	3.64	2.3
CITY-C	3.5	6.4	4.18

这些数据排成一个三行四列的矩阵。矩阵中的元素，事实上可以看成二维向量。每行(或每列)中各向量依次排成一个元串(TUPLE)，称行元串(或列元串)。在m×n矩阵中，行元串的个数等于数行m，行元串中向量的个数等于列数n。当我们用Visicalc的命令/S#S生成DIF文件时，若以列顺序(用命令C)则产生行元串。这样，前导记录中第二、三组中参数v=n，t=m；反之，若以行顺序(用命令R)，则产生元串，v=m，t=n。例如，上述城市降雨表，今以列顺序生成DIF文件，前导记录中的参数v=4，t=3。弄清参数v，t与矩阵表的行、列数的这个关系，对用BASIC程序正确地读、写DIF文件中的数据，至为重要。

(2)　数据记录(Data Record)。

这部分是Visicalc工作数据本身的记录(不包含单元中有关运算的信息)。它按元串的顺序排列。每一元串，都以表示起始的特定标志开头，接着是一个个数据的形态标志与内容，其一般格式为：

```
-1，0   }
BOT     } (元串起始标志)

T1，N1   }
"字串1" } (元串的第一项)

T2，N2   }
"字串2" } (元串的第二项)
 ⋮
Tn，Nn   }
"字串n" } (元串的第n项)
```

这里，起始标志中"BOT"是表示Beginning of tuple

(即元串的起始)。第n项中，Tn是该项数据的形态标示值。标示值有三种：

- −1　表示起始或结束；
- 0　表示是数值型数据；
- 1　表示是字串型数据。

N_n是第n项数值型数据记录本身。当数据是数值时，"字串n"部分被字符V代替，即表示是值(Value)。当数据是字串型时，N_n取0，而"字串n"部分就是记录第n项的字串本身。例如，上述城市降雨量表中第一行元串的数据记录为：

```
-1,0
BOT
1,0
"CITY-A"
0,5.2
V
0,4.33
V
0,5.32
V
```

(3)结束记录(End of Data Record)

结束记录是DIF文件结束的标志，其格式为：

```
-1,0
EOD
```

这里EOD是End of Data(数据结束。)

这样，城市降雨量表若以列顺序存盘，并名为RAIN.DIF，则RAIN.DIF全部记录内容如下：

```
TABLE
0,1
""
VECTORS
0,4
""
TUPLES
0,3
""
DATA
0,0
""
-1,0
BOT
1,0
"CITY-A"
0,5.2
V
0,4.33
V
0,5.32
V
-1,0
BOT
1,0
"CITY-B"
0,4.1
V
0,3.64
V
0,2.3
V
-1,0
BOT
1,0
"CITY-C"
0,3.5
V
0,6.4
V
0,4.18
V
-1,0
EOD
```

三、显示DIF文件

弄清DIF文件整个结构的格式后，就不难用BASIC语言将全部资料读出。下面就是这样的一个APPLESOFT BASIC程序，它读出DIF文件的全部资料并显示于屏幕；对于其他系统，只要修改相应的读顺序型数据文件的语句即可。

```
10  HOME
20  INPUT "NAME OF FILE? ";F$
30  HOME
40  PRINT"* * * DISPLAY";F$;"* * *"
50  PRINT "  "
60  D$=CHR$(4)
70  PRINT D$;"OPEN";F$
80  PRINT D$;"READ";F$
90  Q$=CHR$(34)
100 INPUT T$
110 INPUT V1,V2
120 INPUT S$
130 PRINT T$
140 PRINT V1;",";V2
150 PRINT Q$;S$;Q$
160 IF T$< >"DATA"GOTO 100
170 INPUT V1,V2
180 INPUT S$
190 PRINT V1,",";V2
200 IF V1=1 THEN PRINT Q$;S$;Q$
```

```
210 IF V1 <> 1 THEN PRINT S$
220 IF S$<  >"EOD"GOTO 170
230 PRINT D$;"CLOSE"
240 END
```

这是显示 DIF 文件的通用程序。运行后，只要给出 DIF 文件名，就能显示出任一 DIF 文件的全部资料。其中，70～90 句是开读顺序型数据文件的命令；90 句中，34 是引号（"）的 ASCII 十进代码；100－160 句是读出并显示前导记录；170－220 句是读出并显示数据记录与结束记录；240 句是关闭数据文件命令。

四、数据由 DIF 文件读入 BASIC 变量

显然，DIF 文件的全部资料不都是 Visicalc 工作数据。我们只需全部工作数据读入 BASIC 变量，其他资料可以屏弃。一般来说，读数据的程序，须针对具体的 DIF 文件编写。例如，在 APPLESOF BASIC 下，读 RAIN、DIF 文件的程序为：

```
10  HOME
20  D$=CHR$(4)
30  INPUT "NAME OF FILE ? ";F$
40  PRINT D$;"OPEN";F$
50  PRINT D;"READ";F$
60  INPUT T$
70  INPUT V1,V2
80  INPUT S$
90  IF T$="VECTORS"  THEN V=V2
100 IF T$="TUPLES"  THEN T=V2
110 IF T$<  > "DATA" THEN 60
120 DIM X $(T),X(T,V-1)
130 FOR I=1 TO T
140 INPUT V1,V2
150 INPUT S$
160 FOR J=0 TO V-1
170 INPUT V,V2
180 INPUT S$
190 IF V1=1 THEN  X$(I)=S$
200 IF V1=0 THEN  X(I,J)=V2
210 NEXT J
220 NEXT I
230 PRINT D$;"CLOSE"
260 END
```

这里，60－80 句读 DIF 文件开始标志；90－120 句是从前记录中读出参数 v 和 t，并用它们来确定存贮数据的数组变量的维数；190-200 句判别数据的类型，以便读入不同类型的变量。容易看出程序运行后，各城市的名将读入变量 X$(I)，三个月的降雨量将分别读入变量 X(I,J)。

当数据传送于 BASIC 变量后，若需进一步处理这些数据，可继续用 BASIC 语言编写程序。例如，对城市降雨量表，我们需按各月平均降雨量的大小排序，以建立新表。为此，只要在以上程序中补入排序的子程序：

```
1000 DIM MX(T)
1010 FOR I=1 TO T
1020 FOR J=1 TO V-1
1030 MX(I)=MX(I) +X(I,J)
1040 NEXT J
1050 NEXT I
1060 FOR P=1 TO T-1
1070 FOR Q=P+1 TO T
1080 IF MX(P)>=MX(Q) THEN 1200
1090 W=MX(P)
1100 MX(P)=MX(Q)
1110 MX(Q)=W
1120 W$=X$(P)
1130 X$(P)=X$(Q)
1140 X$(Q)=W$
1150 FOR K=1 TO V-1
1160 W=X(P,K)
1170 X(P,K)=X(Q,K)
1180 X(Q,K)=W
1190 NEXT K
1200 NEXT Q
1210 NEXT P
1220 RETURN
```

五、回送数据于 Visicalc

经 BASIC 语言处理后的数据，最后往往是要送回 Visicalc，这样，才完全利用了它的特性。这步工作，实际上是按要求的格式，将数据写成 DIF 文件。这里，我们仍以降雨量的表为例，将上述经排序处理后的数据，用 APPLESOFT BASIC 编写生成 DIF 文件的子程序。

```
2000 Q$=CHR$(34)
2010 PRINT
2020 INPUT "NAME OF NEW FILE? ";F$
2030 PRINT D$;"OPEN";F$
2040 PRINT D$;"WRITE";F$
2050 V1=0:V2=1
2060 T$="TABLE";S$=" "
2070 GOSUB 2320
2080 T$="VECTORS";V2=V+1
2090 GOSUB 2320
2100 T$="TUPLES";V2=T
2110 GOSUB 2320
2120 T$="DATA";V2=0
2130 GOSUB 2320
2140 FOR I=1 TO T
2150 V1=-1:V2=0 :S$="BOT"
2160 GOSUB 2330
```

（下转 15 页）

在 APPLE CP/M 操作系统中增加文件比较命令 COMP

宁波效实中学　施吉鸣

在 IBM PC/XT MS—DOS 操作系统中，文件操作命令的其中之一是文件比较命令 COMP，这条命令用于文件间的比较，可用来检查文件复制的正确性，通常这条命令可以在 COPY 命令操作之后使用，以确保复制文件和源文件完全相同。然而在 APPLE CP/M 操作系统中却无这个文件比较命令。

最近笔者在 APPLE 联想式汉字系统上(该系统由 APPLE 微机系统和汉字扩展部件两部分组成，在 CP/M 操作系统下，由 Z80 CPU 控制该扩展部件进行工作)对文字编辑软件 Wordstar 加以汉化，在开发过程中，由于需要，在 CP/M 操作系统中增加了文件比较命令 COMP，该命令比较 IBM PC/XT MS—DOS 中的 COMP 显的简单灵活，同时在此基础上又加强了其功能，使得对不同长度的文件亦可进行比较。在比较过程中，如果发现不同，还将显示相应地址单元的不同值。由于 APPLE CP/M 操作系统中文件不采用树形结构目录，因此该命令行格式中省略目录路径这一可选项。

现将该命令的形成及使用方法介绍如下。

一、COMP 的命令生成

操作者将本文后面源程序通过 ED 命令以 COMP、E80 名字存在盘上，然后经汇编(ASMB)、连接(LINK)后可生成 COMP COM 文件，即文件比较命令 COMP。

二、文件比较命令 COMP 的使用方法

功能：比较两个指定的文件内容。

命令行格式：

COMP[d:]filenamei[.ext]　[d:]filename2[.ext]

说明：这条命令是一个外部命令。它进行的是两个文件间的比较，而不是两个整盘内容的比较。在命令行中，第一个参数量是初级文件，第二个参量是次级文件，要进行比较的两个文件可以在相同的盘上，也可以是在不同的盘上。如果命令行中所指定的文件之一在相应的盘上找不到，将显示 File not found.(文件未找到)如果比较中发现有不同的内容，将显示相应的地址及两个不同的内容。如果比较成功，则显示 File compare OK。

使用 COMP 命令应该注意以下两点：

(1) 如果要比较的两个文件是在两个不同的盘上，那么它们可以有相同的文件名。

(2) 如果文件的长度不同，则对这样的两个文件进行比较时，比较工作到较短文件结束时为止。

```
; compare two files, if data in same address is not equal, display;
address and data. if files'length is not equal, compare uneill end of
shorter filie.
BOOT      EQU     0000H
BDOS      EQU     0005H
FCB1      EQU     005CH
FCB1CR    EQU     FCB1+32
FCBO      EQU     006CH
DBUF1     EQU     0080H
TPA       EQU     0100H
PRINT     EQU     9
OPENF     EQU     15
READF     EQU     20
DMAF      EQU     26
          ORG     TPA
START     LD      DE,CPRT
          CALL    DISP
; MOVE    FCB2
          LD      HL,FCBO
          LD      DE,FCB2
          LD      BC,10H
          LDIR
; set counters to primary value
          XOR     A
          LD      (FCB2CR),A
          LD      (FCB1CR),A
          LD      C,A
          LD      B,A
          PUSH    BC
;OPEN     FILES
          LD      DE,FCB1
          CALL    OPEN
          LD      DE,NOFILE
          INC     A
          JR      Z,FINIS
          LD      DE,FCB2
          CALL    OPEN
          LD      DE,NOFILE
          INC     A
          JR      Z,FINIS
;Read files
```

```
LOOP    LD      DE,DBUF1
        CALL    DMA
        LD      DE,FCB1
        CALL    READ
        OR      A
        JR      NZ,EOFILE
        LD      DE,DBUF2
        CALL    DMA
        LD      DE,FCB2
        CALL    READ
        OR      A
        JR      NZ,EOFILE
        POP     BC
        EXX
        LD      C,80H
        LD      DE,DBUF1
        LD      HL,DBUF2
COMP    LD      A,(DE)
        LD      B,(HL)
        CP      B
        JR      Z,OK
        CALL    NOTEQU
OK      DEC     C
        LD      A,C
        OR      A
        JR      Z,READRD
        INC     DE
        INC     HL
        EXX
        INC     BC
        EXX
        JR      COMP
READRD  EXX
        INC     BC
        PUSH    BC
        JR      LOOP
EOFILE  LD      DE,COMPOK

        LD      A,(DIFFER)
        OR      A
        JR      Z,FINIS
        LD      DE,EOCOMP
FINIS   CALL    DISP
        JP      BOOT
; SYSTEM INTERFACE SUBROUTINES
OPEN    LD      C,OPENF
        JP      BDOS
DMA     LD      C,DMAF
        JP      BDOS
READ    LD      C,READF
        JP      BDOS
DISP    LD      C,PRINT
        JP      BDOS
CRLF    LD      E,0DH
        LD      C,2
        CALL    BDOS
        LD      E,0AH
        LD      C,2
        JP      BDOS
NOTEQU  EXX
        PUSH    BC
        EXX
        PUSH    HL
        PUSH    DE
        PUSH    BC
        PUSH    AF
        EXX
        PUSH    BC
        LD      A,B
        CALL    PHEX
        POP     BC
        LD      A,C
        CALL    PHEX
        LD      E,':'
        LD      C,2
        CALL    BDOS
        POP     AF
        CALL    PHEX
        LD      E,' '
        LD      C,2
        CALL    BDOS
        POP     BC
        PUSH    BC
        LD      A,B
        CALL    PHEX
        CALL    CRLF
        LD      A,1
        LD      (DIFFER),A
        POP     BC
        POP     DE
        POP     HL
        EXX
        POP     BC
        EXX
        RET
PHEX    PUSH    AF
        RRC     A
        RRC     A
        RRC     A
        RRC     A
```

```
        CALL    PNIB
        POP     AF
        CALL    PNIB
        RET
PNIB    AND     OFH
        CP      10
        JP      NC,P10
        ADD     '0'
        JR      PRN
P10     ADD     'A'-10
PRN     LD      E,A
        LD      C,2
        CALL    BDOS
        RET
                Data areas
CPRT    DB      0DH,0AH,0AH,'File
                Compare urility',0DH,0AH
        DB      'Copyright by SHI JIMING,
                30—Dec-87',0DH,0AH,0AH,'$'
NOFILE  DB      'File not fond,$'
COMPOK  DB      'Files compareok,$'
EOCOMP  DB      'End of compare,$'
DIFFER  DB      0
FCB2
FCB2CR  EQU     FCB2+32
DBUF2   EQU     FCB2CR+32
        END     SIART
```

令 APPLEII 显示目录命令改为 DIR

中山大学　莫曰华

APPLEII 显示磁盘文件目录的 DOS 命令为 CATALOG，命令名太长，使用起来不方便，而且与其它型号计算机的相应命令不一致，例如 TRS-80、IBM PC 等都是用 DIR。我们可以令 APPLEII 也使用 DIR，方法是将 DOS 在内存中的映象修改，使 DOS 命令名表中的 CATALOG 改为 DIR，然后再生成一片新的 DOS 系统软盘。

CATALOG 在内存命令名表中的首字符地址为 \$A8D2，命令名的 ASCII 码分别为 \$43，\$41，\$54，\$41，\$4C，\$4F，\$C7（最后一字码不是 \$47 而是 \$C7，即最高位加了 1，表示该命令名表完）。现要改为 DIR 即改为 \$44，\$49 和 \$D2，多出的空位置要消除掉。具体做法如下：

先准备两张软磁盘，一片原 DOS3.3 系统盘和另一片空磁盘。

1. 使用原 DOS3.3 系统盘，启动 DOS 系统。
2. 进入监控状态

   ]CALL－151↙
3. 将 \$A8D2— \$A8D4 的内容修改为 \$44，\$49，\$D2
4. 将命令表中 \$A8D9— \$A908 的内容前移四个位置，以消除多出来的空格。

   *A8D5<A8D9. A908M↓
5. 按 CTRL—C 键退出监控，返回 DOS 状态。
6. 插入空磁盘，用 INIT 命令初始化该盘，修改后的 DOS 即存入盘中，成为新系统盘。
7. 将原 DOS3.3 的所有文件用 FID 复制到新系统盘。

至此便完成了显示目录命令的修改，以后用新的系统盘启动，就可以用 DIR 命令来显示磁盘文件目录。

如果想对 DOS 作更多的修改，可参阅由中山大学出版社出版的《APPLEIII 微电脑使用方法与技巧》中有关“DOS 的修改”一节。

APPLEII 机、单板机通用数模转换(A/D)卡

本卡是八位八通道模数转换装置，它能直接与苹果机、Z80 单板机配接，进行数据采集和处理，功能好，价格低。

主要技术指标：

1、八个模拟输入通道，可由程序选择；

2、转换采用 8 位二进制数，并行输出，分辨率为 1/256；

3 转换时间：≤100μS；

4、采数时间：用 BASIC ≤27ms/次

用 6502 汇编 ≤140μs/次；

5、模拟输入电压范围：0—5 伏；

6、工作电流：30mA；

7、外形尺寸：80×110mm。

销售价格为 235 元，需要者请与本刊联系。

在 Applesoft 下超千万数值运算子程序

阜新矿业学院　于建华　阜新市通讯电缆厂　甘重

我们在苹果机上开发财务管理系统时，由于该软件使用了 Applesoft 及仓颉汉卡，遇到超千万数值的运算问题。

Applesoft 的数字变量的有效数字位数只有九位，当位数再多时也只以科学记数法记数。在财务管理系统中不能采用科学记数法来记数，所有的金额数字均须保留小数点后两位有效数字。这样，整数位数最多只有七位。即当金额超过千万以后，分值的金额就自动舍弃为零。如果不能正确地处理有效数字的问题，该财务系统就没有使用价值。为此，我们编写了一个专用子程序。它可以进行两个数字的加减法。其数字的整数位数不限，均能正确运行，并精确到小数点后两位，满足了财务管理系统的需要。解决此问题对推广苹果机的使用还是有价值的。因此，我们将此程序提供给读者参考。

一、使用要求及注意事项：

(1)输入：

①输入的两个数均须是字符变量。

②输入数字的标准格式为：

…×××，×××或…×××.×0 或…×××.00 即整数位数不限，而小数点后必须是两位字符。如果是零时，也要将零字符输入。

③正数可以不带符号输入。负数必须带“—”号输入。

④以加法为例：调用该子程序前，须将加数送给变量 FIRST$；被加数送给变量 SEC$。

(2)输出：

①所得结果(和或差)存放在变量 SG$中。

②输出的结果符合输入的“标准形式”。即小数点后两位已经是自动添零补齐。所以如果做连续的加减法运算，就不用考虑是否符合其标准形式。其结果可以直接打印。符合财务要求的形式：即小数点后两位自动添零补齐。

(3) 注意该程序使用的变量名较多。因此与主程序中变量名不要发生冲突，以免出错。

该程序的优点是：当运行时，如果两个数字均不超过百万时，其运行速度不受影响。一旦其中一个数字超过百万时，特别是做大量连续加减法时，频繁调用该子程序时才会降低其运行速度。但仍不失其使用价值。

(4) 该程序是以两数做加法为例。如欲做减法时，将输入的减数的符号改变之后再调用该子程序即可。

(5)该子程序用一个自定义函数：

FNA(DV)＝INT(DV＊100＋0.5)/100 它用于四舍五入以保证精确到小数点后两位。应将其先写入主程序内。

二、程序

```
2500 A1=VAL (FIRST$):B1=VAL (SEC$)
2502 IF ABS (A1)>999999.99 OR ABS (B1)>999999.99 THEN 2506
2504 C1=A1+B1:DV=C1:DV=FN A(C1)
2505 GOSUB 3000:RETURN
2506 AA$=FIRST$ :GOSUB 2600:B1$=Q$ :BVS=QV$
2507 AA$=SEC$:A1=B1:GOSUB 2600
2508 C1=VAL(B1$)+VAL(Q$):DV=VAL(BV$)+VAL(QV$):DV=FN A(DV)
2510 IF C1< >O THEN 2514
2512 GOSUB 2580:RETURN
2514 IF DV=O THEN GOSUB 2800:RETURN
2516 IF C1<O THEN 2530
2518 IF DV<O THEN 2524
2520 IF DV>=1000000 THEN C1=C1+1:DV=DV-1000000:DV=FN A(DV)
2522 GOSUB 2580:RETURN
2524 C1=C1-1:DV=DV+1000000:DV=FN A(DV):GOSUB 2580:RETURN
2530 IF DV<0 THEN 2534
```

```
2532 C1=C1+1:DV=DV-1000000
     :DV=ABS(DV):GOSUB 2580
     :RETURN
2534 DV= ABS (DV)
2536 IF DV> =1000000 THEN C1.
     C1-1:DV=DV-1000000:DV=
     FN A(DV)
2538 GOSUB 2580 :RETURN
     IF C1=0 THEN GOSUB 3000:
     RETUTN
2582 GOSUB 2800:RETURN
2600 AB= ABS(A1):IF AB> =1
     .000000 THEN 2620
2610 Q$="0":QV$=AA$:RETURN
2620 QV$= RIGHT$ (AA$,9):D= LEN
     (AA$)
2630 DA = LEN (QV$):D=D - DA:Q
     $= LEFT$ (AA$,D)
2640 IF A1>0 THEN RETURN
2650 QV$="-"+QV$:RETURN
2652 REM---------
2800 C1$= STR$ (C1):IF DV= 0 THEN
     SG$=C1$+"000000.00":RETURN
2810 IF DV<0 THEN 2830
2820 GOSUB 2850:SG$=C1$+SG$:
     RETURN
2830 DV = ABS (DV):GOTO 2820
2850 DZ$="O":DV$= STR$ (DV *
     100):DV$= RIGHT$ (DV$,2):D
     = LEN (DV$):IF D<1 THEN
     DV$=("O"+DV$
2852CI$ - STR$ ( INT (DV)):D =
     LEN (CI$):D=6-D
2855 IF D =0 THEN SG$=CI$+"
     ."+DV$:RETURN
2858 IF D =1 THEN 2865
2860 FOR X =1 TO D-1:DZ$= DZ
     $+"0":NEXT X
2865 SG$ = DZ$+CI$+"."+DV$:
     RETURN
2868 REM-------------
3000 IF DV =0 THEN SG$="0.00"
     :RETURN
3010 IF DV <0 THEN 3030
3015 IF DV <1 THEN SG$="0"+
     STR$(DV):RETURN
3020 GOSUB 3050:RETURN
3030 DV = ABS (DV):GOSUB 3050:IF
     DV<1 THEN SG$="-0"+SG$
     :RETURN
3040 SG$="-"+SG$:RETURN
3050 DV$= STR$(DV * 100):DV$=
     RIGHT$ (DV$.2):CI$= STR$
     (INT (DV)):SG$=CI$+"."+
     DV$:RETURN
```

(上接 38 页)

```
170  Y=Y+1: IF INT (Y/ (X2-X1)) <=Y2
     AND INT (Y/ (X2-X1)) >=Y1 AND P=1
     THEN PRINT A$;
175  IF P=2 THEN A$(Y)=A$
180  NEXT: IF INT (Y/ (X2-X1)) <=Y2 AND
     INT (Y/ (X2-X1))) =Y1 AND P=1 THEN
     PRINT"! ":PRINT"! ";
190  NEXT: NEXT: IF P=1 THEN NEXT: GOTO
     240
200  FOR E=1 TO 7: PRINT "! ";: FOR F=Y2
     TO Y1 STEP -1
210  PRINT MID$ (A$(F),E,1);
220  NEXT F:PRINT"! ":NEXT E
230  NEXT
240  PRINT:PRINT:PRINT:PRINT
245  PR#0
250  END
500  PR# 1
505  PRINT:PRINT:PRINT:PRINT
510  IF P=1 THEN PP =5:GOTO 520
515  PP=4
520  PRINT CHR$(27);"A";CHR$(PP)
525  PRINT CHR$(15)
526  PRINT CHR$(27);"G"
530  RETURN
```

大批量数据的快速排序

南通师专　宋　岳

排序在实际应用中是经常碰到的，尤其对批量大的排序，运行速度快慢与消耗内存多少是至关紧要的。

若是对几十个或几百个数据的排序，即使效率低些，问题还不大，当数据超过一千以上甚至一万以上时，其排序方法的优缺点，往往成了工作的成败关键之一，有的运行速度高，但占有内存大。有的内存节省了，却效率低，要做到既要速度快，又要省内存，不太容易。对已介绍的排序，确实有不少好的，但都难于照顾到高效、低耗的两全之策。不论是冒泡法、对分法……都离不开大小比较和数据交换这个思路，因此效率和内存占有方面总是改进不大。

我们在实践中摸索到了一种方法。打破常规，不比较、不交换，数据进入内存一次排序成功。也就是说最后一个数据进入内存，排序就宣告已完成。这是在排序上的一种突破，一般的方法在苹果机上对一万个数据排序至少要在六十小时以上。而且我们这个方法，用不了一个小时，而这一小时不全是用在排序上，而是包含产生数据，上边讲过排序是几乎不占时间的，只要最后一个数进入内存，排序就已宣告完成了。我们这种排序的方法，经过多次检验是可行的，结合不同的实际应用，单独编制一此程序效率会更高，现将苹果机上已试验结果说明如下：

数据个数	运行时间
1000	0.5 分钟
2000	1　分钟
3000	2　分钟
4000	3　分钟

以上时间包括产生数据，实际排序时还要分一般批量大的数据不在苹果机上排序，据最近介绍比较好的程序。在苹果机上处理的最多数据是 1200 个，若是 1400 个数据，就因为内存不是而不能运行了。本程序可以处理两万个数据，假如对程序捎作补充和修改，就可以在苹果机上处理上百万个数据的排序。

程序说明：

以招生录取为例，假定考试六门课程总分为 600。

一、确定数组元素：30 句中 A 数组中存放不同的总分数。如 A(529)就表示总分为 529 的全部人数，在程序执行时对具有不同总分的人数就分别累加到对应的数组元素中，在 A$字符串数组中存放不同总分的考生号码(或名字)如 A$(529)表示所有总分为 529 的考生号码(或名字)，全部以字符形式存放在 A$(529)中。

二、利用数组元素的有序性，实现数据排序，在程序执行时，对具有不同总分的人数和考生号码就已分别进入对应的数组元素与字符数组元素，所以实际上已实现了排序。

三、一共只用了两个单循环，第一个循环实现读数与排序。第二个循环打印输出。整个程序只有十五行，在苹果机上可以直接处理二万个数据的排序，运行时间却只有同类排序时间的五十分之一。

```
]LIST
10  INPUT"N=";N:N=N+1000
20  PRINT "NO:"; TAB (10) "CS:"; TAB (20)
    "TO:"
30  DIM A(600),A$(600)
40  FOR I=1001 TO N
50  S=INT(RND(1) * 600)
60  A(S)=A(S)+1
70  A$(S)=A$(S)+STR$(I)
80  NEXT I:T=0
90  FOR I=599 TO 1 STEP -1
100  IF A(I)=0 THEN 150
110  FOR S=0 TO A(I) -1
120  T=T+1
130  PRINT T; TAB( 10); MID$ (A$(I),1+S *
     4,4);TAB(20);I
140  NEXT S
150  NEXT I
160  END
```

```
]LIST
10  INPUT"N=";N:N=N+1000
20  PRINT "NO:"; TAB (10) "CS:"; TAB (20)
    "TO:"
30  DIM A(600),A$(600)
40  FOR I=1001 TO N
50  S= INT(RND(1) * 600)
60  A(S)=A(S)+1
70  A$(S)=A$(S)+ STR$(I)
80  NEXT I:T=0
90  FOR I=599 TO 1 STEP -1
```

```
100  IF A(I)=0 THEN 150
110  FOR S=0 TO A(I)-1
120  T=T+1
130  PRINT T; TAB(10); MID$(A$(I),1+S*4,
     4);TAB(20);I
140  NEXT S
150  NEXT I
160  END
]RUN
N=10
NO:      CS      TO:
```

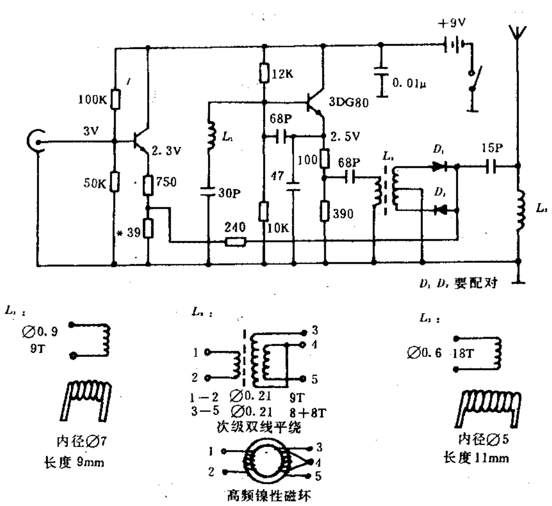
1	1008	527
2	1003	507
3	1004	504
4	1002	485
5	1005	355
6	1007	251
7	1009	221
8	1006	160
9	1001	78
10	1010	73

·经验交流·

87 年第 1 期《问题征答》参考答案

上海科技大学无线电系　　潘锐

如果想用黑白或彩色电视机作为显示器，可将 Apple 机的视频信号输出端电缆直接输入至下图调制器。调制器的发射频率可在 1～5 频道之间任意调整。该调制器至少有 10 米的发射距离，使用十分方便，也不影响电视机的正常收看。调试方便，只需将各工作点调好后，拉动 L_1 线圈可调整至二频道。直至图象稳定为止。



调　试　程　序

浙江大学中心实验室　郑　平

有一个功能较强的调试软件对加快编制和分析机器语言程序是很有效的。在早期苹果机监控程序中，除有内存、寄存器检查修改和反汇编等指令外，还有单步跟踪(Step)和连续跟踪(Trace)功能。但是现在多数机器没有此功能。本文根据原非自动启动监控程序中的 S、T 指令进行改编、增设了可设断点的跟踪指令。

方法一：因为当使用语言板上整数 BASIC 的同时，监控程序也是使用语言板上的。所以如果有语言板，我们就可以直接在语言板上修改监控程序，使之具有 S(单步跟踪)和 T(连续跟踪)指令的功能。改写方法：将 INTBASIC 装入语言板后，从整数 BASIC 进入监控程序，键入：

```
*C081 <CR>
C083 <CR>
```

解除语言板的写保护，键入下面列出的内存值。

```
*FA40.FB18
FA40— FF FF FF 20 D0 F8 68 85
FA48— 2C 68 85 2D A2 08 BD 10
FA50— FB 95 3C CA D0 F8 A1 3A
FA58— F0 42 A4 2F C9 20 F0 59
FA60—C9 60 F0 45 C9 4C F0 5C
FA68— C9 6C F0 59 C9 40 F0 35
FA70— 29 1F 49 14 C9 04 F0 02
FA78— B1 3A 99 3C 00 88 10 F8
FA80— 20 3F FF 4C 3C 00 85 45
FA88— 68 48 0A 0A 0A 30 03 6C
FA90— FE 03 28 20 4C FF 68 85
FA98— 3A 68 85 3B 20 82 F8 20
```

```
FAA0— DA FA 4C 65 FF 18 68 85
FAA8— 48 68 85 3A 68 85 3B A5
FAB0— 2F 20 56 F9 84 3B 18 90
FAB8— 14 18 20 54 F9 AA 98 48
FAC0— 8A 48 A0 02 18 B1 3A AA
FAC8— 88 B1 3A 86 3B 85 3A B0
FAD0— F3 A5 2D 48 A5 2C 48 20
FAD8— 8E FD A9 45 85 40 A9 00
FAE0— 85 41 A2 FB A9 A0 20 ED
FAE8— FD BD 1E FA 20 ED FD A9
FAF0— BD 20 ED FD B5 4A 20 DA
FAF8— FD E8 30 E8 60 18 A0 01
FB00— B1 3A 20 56 F9 85 3A 98
FB08— 38 B0 A2 20 4A FF 38 B0
FB10— 9E EA EA 4C 0B FB 4C FD
FB18— FA
*FB9D

FB9D— EA
*FBB3.FBBA

FBB3—C4 C2 C1 FF C3
FBB8—FF FF FF
*FEC2.FEC9

FEC2—C6 34 20 75 FE 4C
FEC8—43 FA
*FFCF

FFCF—ED
*FFD2

FFD2—EC
*FFE9

FFE9—C3
*FFFC.FFFE

FFFC—59 FF 86
```

修改完后把 \$D000—\$FFFF 的内容以 INTBASIC 名存入磁盘，便可使用 S，T 指令。命令形式：

1、地址 A_1 S　　以 PC= A_1 跟踪一步；

2、　S　　　继续跟踪一步；

3、地址 A_1 T　　以 PC= A_1 连续跟踪。

方法二：在方法一的基础上，利用 CTRL—Y 用户指令增加可设断点的跟踪指令。为了防止用户程序与该程序重选，在程序中都用相对转移指令。这样，只要把 3F8 地址的转移指令指向调试程序的起始地址，就可以把这段程序移到任意一段自由 RAM 中。程序为：

```
*3F8L
03F8—  4C  00  10  JMP  $1000
1000—  A5  3E      LDA  $3E
1002—  85  02      STA  $02
1004—  A5  3F      LDA  $3F
1006—  85  03      STA  $03
1008—  A9  00      LDA  #$00
100A—  85  01      STA  $01
100C—  A9  01      LDA  #$01
100E—  85  00      STA  $00
1010—  8A          TXA
1011—  F0  1E      BEQ  $1031
1013—  A5  42      LDA  $42
1015—  C5  44      CMP  $44
1017—  D0  24      BNE  $103D
1019—  A5  43      LDA  $43
101B—  C5  45      CMP  $45
101D—  D0  1E      BNE  $103D
101F—  85  3D      STA  $3D
1021—  A5  42      LDA  $42
1023—  85  3C      STA  $3C
1025—  A5  3A      LDA  $3A
1027—  C5  02      CMP  $02
1029—  D0  0A      BNE  $1035
102B—  A5  3B      LDA  $3B
102D—  C5  03      CMP  $03
102F—  D0  04      BNE  $1035
1031—  A9  FF      LDA  #$FF
1033—  85  01      STA  $01
1035—  20  C4  FE  JSR  $FEC4
1038—  A5  01      LDA  $01
103A—  FO  E9      BEQ  $1025
103C—  60          RTS
103D—  38          SEC
103E—  A5  02      LDA  $02
1040—  E9  01      SBC  #$01
1042—  85  02      STA  $02
1044—  A5  03      LDA  $03
1046—  E9  00      SBC  #$00
1048—  85  03      STA  $03
104A—  90  12      BCC  $105E
104C—  20  C4  FE  JSR  $FEC4
104F—  A5  02      LDA  $02
1051—  38          SEC
1052—  E9  01      SBC  #$01
1054—  85  02      STA  $02
1056—  A5  03      LDA  $03
1508—  E9  00      SBC  #$00
105A—  85  03      STA  $03
105C—  B0  EE      BCS  $104C
```

```
105E—    60                 RTS
```

方法三:没有语言板或不改写监控程序,可以使用下面的程序:

```
*3F8L

03F8—   4C   00   10   JMP   $1000

1000—   A5   3E        LDA   $3E
1002—   85   02        STA   $02
1004—   A5   3F        LDA   $3F
1006—   85   03        STA   $03
1008—   A9   00        LDA   #$00
100A—   85   01        STA   $01
100C—   8A             TXA
100D—   F0   1E        BEQ   $102D
100F—   A5   42        LDA   $42
1011—   C5   44        CMP   $44
1013—   D0   24        BNE   $1039
1015—   A5   43        LDA   $43
1017—   C5   45        CMP   $45
1019—   D0   1E        BNE   $1039
101B—   85   3D        STA   $3D
101D—   A5   42        LDA   $42
101F—   85   3C        STA   $3C
1021—   A5   3A        LDA   $3A
1023—   C5   02        CMP   $02
1025—   D0   0A        BNE   $1031
1027—   A5   3B        LDA   $3B
1029—   C5   03        CMP   $03
102B—   D0   04        BNE   $1031
102D—   A9   FF        LDA   #$FF
102F—   85   01        STA   $01
1031—   20   5B   10   JSR   $105B
1034—   A5   01        LDA   $01
1036—   F0   E9        BEQ   $1021
1038—   60             RTS
1039—   38             SEC
103A—   A5   02        LDA   $02
103C—   E9   01        SBC   #$01
103E—   85   02        STA   $02
1040—   A5   03        LDA   $03
1042—   E9   00        SBC   #$00
1044—   85   03        STA   $03
1046—   90   12        BCC   $105A
1048—   20   5B   10   JSR   $105B
104B—   A5   02        LDA   $02
104D—   38             SEC
104E—   E9   01        SBC   #$01
1050—   85   02        STA   $02
1052—   A5   03        LDA   $03
1054—   E9   00        SBC   #$00
1056—   85   03        STA   $03
1058—   B0   EE        BCS   $1048
105A—   60             RTS
105B—   20   75   FE   JSR   $FE75
105E—   20   D0   F8   JSR   $F8D0
1061—   68             PLA
1062—   85   2C        STA   $2C
1064—   68             PLA
1065—   85   2D        STA   $2D
1067—   A2   08        LDX   #$08
1069—   BD   2B   11   LDA   $112B,X
106C—   95   3C        STA   $3C,X
106E—   CA             DEX
106F—   D0   F8        BNE   $1069
1071—   A1   3A        LDA   ($3A,X)
1073—   F0   42        BEQ   $10B7
1075—   A4   2F        LDY   $2F
1077—   C9   20        CMP   #$20
1079—   F0   59        BEQ   $10D4
107B—   C9   60        CMP   #$60
107D—   F0   45        BEQ   $10C4
107F—   C9   4C        CMP   #$4C
11081—  F0   5C        BEQ   $10DF
1083—   C9   6C        CMP   #$6C
1085—   F0   59        BEQ   $10E0
1087—   C9   40        CMP   #$40
1089—   F0   35        BEQ   10C0
108B—   29   1F        AND   #$1F
108D—   49   14        EOR   #$14
108F—   C9   04        CMP   #$04
1091—   F0   02        BEQ   $1095
1093—   B1   3A        LDA   ($3A),Y
1095—   99   3C   00   STA   $003C,Y
1098—   88             DEY
1099—   10   F8        BPL   $1093
109B—   20   3F   FF   JSR   $FF3F
109E—   4C   3C   00   JMP   $003C
10A1—   85   45        STA   $45
10A3—   68             PLA
10A4—   48             PHA
10A5—   0A             ASL
10A6—   0A             ASL
10A7—   0A             ASL
10A8—   30   03        BMI   $10AD
10AA—   6C   FE   03   JMP   ($03FE)
10AD—   28             PLP
10AE—   20   4C   FF   JSR   $FF4C
10B1—   68             PLA
```

```
10B2-   85 3A       STA  $3A
10B4-   68          PLA
10B5-   85 3B       STA  $3B
10B7-   20 82 F8    JSR  $F882
10BA-   20 F5 10    JSP  $10F5
10BD-   4C 65 FF    JMP  $FF65
10C0-   18          CLC
10C1-   68          PLA
10C2-   85 48       STA  $48
10C4-   68          PLA
10C5-   85 3A       STA  $3A
10C7-   68          PLA
10C8-   85 3B       STA  $3B
10CA-   A5 2F       LDA  $2F
10CC-   20 56 F9    JSR  $F956
10CF-   84 3B       STY  $3B
10D1-   18          CLC
10D2-   90 14       BCC  $10E8
10D4-   18          CLC
10D5-   20 54 F9    JSR  $F954
10D8-   AA          TAX
10D9-   98          TYA
10DA-   48          PHA
10DB-   8A          TXA
10DC-   48          PHA
10DD-   A0 02       LDY  #$02
10DF-   18          CLC
10E0-   B1 3A       LDA  ($3A),Y
10E2-   AA          TAX
10E3-   88          DEY
10E4-   B1 3A       LDA  ($3A,)Y
10E6-   86 3B       STX  $3B
10E8-   85 3A       STA  $3A
10EA-   B0 F3       BCS  $10DF
10EC-   A5 2D       LDA  $2D
10EE-   48          PHA
10EF-   A5 2C       LDA  $2C
10F1-   48          PHA
10F2-   20 8E FD    JSR  $FD8E
10F5-   A9 45       LDA  #$45
10F7-   85 40       STA  $40
10F9-   A9 00       LDA  #$00
10FB-   85 41       STA  $41
10FD-   A2 FB       LDX  #$FB
10FF-   A9 A0       LDA  #$A0
1101-   20 ED FD    JSR  $FDED
1104-   BD 39 10    LDA  $1039。X
1107-   20 ED FD    JSR  $FDED
110A-   A9 BD       LDA  #$BD
110C-   20 ED FD    JSR  $FDED
110F-   B5 4A       LDA  $4A,X
1111-   20 DA FD    JSR  $FDDA
1114-   E8          INX
1115-   30 E8       BMI  $10FF
1117-   60          RTS
1118-   18          CLC
1119-   A0 01       LDY  #$01
111B-   B1 3A       LDA  ($3A),Y
111D-   20 56 F9    JSR  $F956
1120-   85 3A       STA  $3A
1122-   98          TYA
1123-   38          SEC
1124-   B0 A2       BCS  $10C8
1126-   20 4A FF    JSR  $FF4A
1129-   38          SEC
112A-   BO 9E       BCS  $10CA
112C-   EA          NOP
112D-   EA          NOP
112E-   4C 26 11    JMP  $1126
1131-   4C 18 11    JMP  $1118
1134-   C1 D8
1136-   D9 D0 D3
```

下面是用BASIC编写的引导程序它可将以NEWTRACE命名的上述机器语言程序移到任一段内存中,其后的使用方法与方法二相同。

```
10  HOME:HTAB 8:VTAB 8
20  PRINT"What start address? "
30  A=4096
40  PRINT:PRINT
50  PRINT:PRINT"Now it is 4096($1000-$1138)";
60  INPUT A$
70  IF A$<  >""THEN A=VAL(A$)
80  PRINT CHR$(4);"BLOAD NEWTRACE,A";A
90  POKE 1016,76:POKE 1017,A-256 * INT(A/256):
    POKE 1018,INT(A/256)
100 AL=A+91-256 * INT ((A+91) /256): AH=
    INT((A+91)/256)
110 POKE A+50,AL:POKE A+73,AL
120 POKE A+51,AH:POKE A+74,AH
130 POKE A+187, A+245-256 * INT ((A+245) /
    256):POKE A+188,INT((A+245)/256)
140 POKE A+106, A+299-256 * INT ((A+299) /
    256):POKE A+107,INT ((A+299)/256)
150 POKE A+261,A+57-256 * INT((A+57)/256):
    POKE A+262,INT ((A+57)/256)
160 POKE A+303, A+294-256 * INT ((A+294) /
    256):POKE A+304,INT((A+294)/256)
170 POKE A+306, A+280-256 * INT ((A+280) /
    256):POKE A+307,INT((A+280)/256)
```

让 Apple 奏曲子

傅增明

下面有两个程序：其一是用简谱把曲子写入计算机，并让计算机奏出曲子，如果需要，可以把曲子写入软盘；其二是用来演奏已在软盘中的曲子。

程序一：10——40 号语句输入音乐子程序。52—65 号语句，用来确定 1=C，D，E，F，G，A，B，75—84 号语句，确定奏曲速度。90—150 语句，输入曲子的简谱符号。160—400 号语句，将简谱代号翻译成音乐数据。410—500 号语句奏曲。520—790 号语句将曲子存入磁盘，其中 550—620 号语句写曲子内容。630—790 号语句是目录管理程序。

用简谱写曲子示例。H 表示高音，L 表示低音。将程序一装入内存，然后键入

RUN<CR>

ANY KEY TO BEGIN

按任一键，显示：

＃1…… …… ……→

＃＃＃

键入：1L 2L 3L 4L 5L 6L 7L 1 2 3 4 5 6 7 1H <CR>

GO ON(Y/N)键入“Y”<CR>

＃2…… …… …… …→

＃＃＃

键入：2H 3H 4H 5H 6H 7H 1HH 2HH 3HH 4HH <CR>

GO ON (Y/N) 键入“Y”<CR>

＃3…… …… …… ……→

＃＃＃

键入：1/2/3/1//2//3//1.2.1L<CR>

GO ON(Y/N)　键入“N”<CR>

微机开始翻译，约 10 秒钟后，Apple 奏曲：

1234567 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 1 2 3 1 2 3 1. 2. !

PUT INTO DISK(Y/N) 键入“Y”<CR>

WHICH NAME 键入 SONG TEXT<CR>

约 20 秒钟，微机将该数据存入磁盘，最后显示：

(L)ISON. (E)ND

按“E”键退出本程序，按“L”键重新奏曲。

程序二：10—40 号语句送入音乐子程序。

45—61 号语句取文件目录。112—145 号语句是屏幕设计。146—210 号语句从磁盘读入歌曲的数据。220—260 号语句奏曲，并提供重听、听另一曲子和退出本程序的选择。

```
4 HOME: VTAB 10: HTAB 5: PRINT "PRESS ANY KEY
  TO BEGIN. ";GET B$
10 FOR I =0 TO 20:READ N:POKE I+770,N:NEXT
40 DATA 173,48,192,136,208,5,206,1,3,240,9,
   202,208,245,174,0,3,76,2,3,96
50 DIM A$(20),F%(500),D%(500),N(14)
52 FOR I=1 TO 14:READ N(I):NEXT
56 DATA 192,172,152,144,128,114,102,96,86,76,
   72,64,57,51
57 INPUT"BASIC NOTE <C.D.E.F.G.A.B>1=";N$
58 IF N$="C" THEN R=1
59 IF N$="D" THEN R=2
60 IF N$="E" THEN R=3
61 IF N$="F" THEN R=4
62 IF N$="G" THEN R=5
63 IF N$="A" THEN R=6
64 IF N$="B" THEN R=7
65 FOR I=R TO R+6:M(I-R+1)=N(I):NEXT
75 PRINT:PRINT:PRINT"  SPEED":PRINT:R=1.02
78 INPUT"(F)AST. (M)ID. (S)LOW  ";N$
80 IF N$="F" THEN R=.8
82 IF N$="M" THEN R=1.02
84 IF N$="S" THEN R=1.25
90 I=0:HOME
100 FOR G=1 TO 20
110 PRINT"#";G;TAB(4);"...... ...... ...... ......>"
120 PRINT"#";: INPUT A$(G)
130 PRINT"GO ON (Y/N)":GET B$
140 IF B$="N" THEN 160
150 NEXT
160 FOR L=1 TO G
170 PRINT
180 PRINT L;TAB( 3);"*";A$(L)
190 PRINT "###";
200 B$=A$(L)
210 FOR K=1 TO 36
220 C$= MID$(B$,K,1)
230 IF C$="1"THEN I=I+1:F%(I)=M(1):D%(I)
    =160:GOTO 380
240 IF C$="2"THEN I=I+1:F%(I)=M(2):D%(I)
    =160:GOTO 380
```

```
250  IF C$="3"THEN I=I+1:F%(I)=M(3):D%(I)
     =160:GOTO 380
260  IF C$="4"THEN I=I+1:F%(I)=M(4):D%(I)
     =160:GOTO 380
270  IF C$="5"THEN I=I+1:F%(I)=M(5):D%(I)
     =160:GOTO 380
280  IF C$="6"THEN I=I+1:F%(I)=M(6):D%(I)
     =160:GOTO 380
290  IF C$="7"THEN I=I+1:F%(I)=M(7):D%(I)
     =160:GOTO 380
300  IF C$="O"THEN I=I+1:F%(I)=1:D%(I)=
     110:GOTO 380
310  IF C$="H"THEN F%(I)=F%(I)/2:GOTO 380
320  IF C$="L"THEN F%(I)=F%(I)*2:GOTO 380
330  IF C$=". "THEN D%(I)=D%(I)*1.56:GOTO0
340  IF C$="/"THEN D%(I)=D%(I)/2.3:GOTO 380
350  IF C$="-"THEN I=I+1:D%(I)=140:F%(I)
     =F%(I-1):GOTO 380
360  IF C$="="THEN 410
370  PRINT"-";:GOTO 390
380  PRINT C$;
390  NEXT K
400  NEXT L
410  FOR L=1 TO I
415  D%(L)=D%(L)*R
420  IF D%(L)<0 THEN D%(L)=0
430  IF D%(L)>255 THEN D%(L)=255
440  IF F%(L)<0 THEN F%(L)=0
450  IF F%(L)>255 THEN F%(L)=255
460  NEXT L
470  FOR L=1 TO I
480  POKE 768,F%(L):POKE 769,D%(L)
490  CALL 770
500  NEXT L
510  PRINT
520  PRINT"PUT INTO DISK (Y/N)";:INPUT B$
530  IF B$="Y"THEN 550
540  GOTO 750
550  INPUT"WHICH NAME";B$
555  ONERR GOTO 790
560  D$=CHR$(4)
570  PRINT D$;"OPEN";B$
580  PRINT D$;"WRITE";B$
590  FOR L=1 TO I +1
600  PRINT F%(L);",";D%(L)
610  NEXT
620  PRINT D$;"CLOSE";B$
630  PRINT D$;"LOCK";B$
635  PRINT D$"UNLOCK";"CATALG#"
640  PRINT D$;"OPEN";"CATALG#"
650  PRINT D$;"READ";"CATALG#"
660  INPUT I
665  DIM C$(I+1)
670  FOR L=1 TO I
680  INPUT C$(L):NEXT
685  PRINT D$;"CLOSE";"CATALG#"
690  C$(I+1)=B$:I=I+1
695  PRINT D$;"OPEN";"CATALG#"
700  PRINT D$;"WRITE";"CATALG#"
710  PRINT I
720  FOR L=1 TO I
730  PRINT C$(L):NEXT
740  PRINT D$;"CLOSE";"CATALG#"
745  PRINT D$;"LOCK";"CATALG#"
750  INPUT"(L) ISON. (E)ND";B$
760 IF B$="L" THEN 470
770  IF B$="E" THEN END
780  GOTO 750
790  I=0:GOTO 690
```

```
10  FOR I=0 TO 20:READ N
20  POKE I+770,N
30  NEXT
40  DATA 173,48,192,136,208,5,206,1,3,240,9,
    202,208,245,174,0,3,76,2,3,96
45  D$=CHR$ (4)
47  PRINT CHR$(4);"OPEN CATALG#"
49  PRINT D$;"ERAD";"CATALG#"
51  INPUT I
53  DIM A$(I)
55  FOR L=1 TO I
57  INPUT A$(L):NEXT
59  PRINT D$;"CLOSE";"CATALG#"
61  PRINT D$
80  DIM F%(450),D%(450)
100  HOME:VTAB 12
110  PRINT TAB(7)"ANYKEY TO BEGIN":PRINT:GET
     A$
112  VTAB 16
113  FOR T=1 TO I
116  PRINT A$(T);":";
117  NEXT
118  PRINT
119  VTAB 1
120  PRINT "* * * * * * * * * * SONG SINGING
     POGM * * * * * * * * * *"
121  PRINT"*<><><><><=========
     ===========<><><><><
```

```
122 PRINT"*>";TAB(14);"SONG PROGRAN——:
    ——FZM";TAB(38);">*":PRINT"*<";TAB
    (24);"-1985.12.8";TAB(38);"<*":PRINT"*
    >
                    >*"
125 PRINT "*<::::  # MUSIC #
    ::::<*":PRINT"*>     >*"
128 PRINT "*>   >*":PRINT"*> )*"
129 PRINT"  *<    <*" PRINT"  *<    <*":
    PRINT"<*::::::::::<*"
130 PRINT"*<><><><><><><><><><
    ><><><*":PRINT"**********
    *****"
135 VTAB 9:HTAB 12:INPUT"WHICH NAME";A$
140 VTAB 9:HTAB 12:PRINT A$;TAB(38);">*":
    VTAB 16
141 FOR T=1 TO I
142 PRINT A$(T);":";
143 NEXT
144 PRINT
145 VTAB 22
146 I=1
148 D$=CHR$(4)
150 PRINT D$;"OPEN";A$
160 PRINT D$;"READ";A$
170 INPUT F%(I),D%(I)
180 IF F%(I)=0 AND D%(I)=0 THEN 200
190 I=I+1:GOTO 170
200 PRINT D$;"CLOSE";A$
210 PRINT D$
220 FOR J=1 TO I-1:POKE 768 F%(J):POKE 769,
    D%(J):CALL 770:NEXT
225 VTAB 23
230 PRINT"(L)ISON (E)ND (N)EXT":GET A$
240 IF A$="L" THEN 220
250 IF A$="E" THEN END
255 IF A$="N" THEN CLEAR:GOTO 10
260 GOTO 230
```

一种更为简单的中文自动启动程序

湖南教育学院计算机室　　杨银洲

在 APPLE—II 机中采用 C—PLUS 汉字卡后，给应用程序中使用汉字带来很多的方便。但多次调用中文自动启动程序（程序一），则在屏幕上多次出现"发明、专利"等字样，这势必影响了应用程序的运行速度并造成了屏幕显示的紊乱。为此，我们采取如下两种办法，克服了原中文自动启动程序的不足之处。

（一）、在应用程序中，第一次所调用的中文自动启动程序，仍采用程序一，荧光屏上相应地显示一次"发明、专利"等字样，但此后在中、西文交替使用中，再次调用中文自动启动程序时，只需将程序一的 20 句内容改为 POKE37984,244 就不会产生"发明、专利"字样。

（二）、应用程序中任何时候调用程序二，均不会显示"发明、专利"等字样，而快速进入中文状态，加速程序运行。

程序一：

```
10   HIMEM:36864
20   POKE  37984,0
30   PR#n
40   PRINT"  "
50   POKE  43603,3
60   POKE  43604,192+n
70   POKE  43605,48
80   POKE  43606,192+n
90   FOR  I=1 TO 15
100  GET  A$
110  NEXT  I
```

程序二：

```
10   HIMEM:36864
20   POKE  37984,244
30   POKE  37985,249
40   POKE  37986,232
50   POKE  37987,227
60   POKE  37988,255
70   POKE  37989,0
80   PR#  n
90   PRINT  "  "
```

APPLE—II 的表格打印

杭州弹黄厂　　雷　蕾

大多数苹果机用户在 DOS3.3 操作系统下用 BASIC 编程时，表格中的横线总是用减号"—"表示，竖线用感叹号"！"表示，画的是虚线和点划线。用这种方法制作的表格不够美观。为了解决这个问题，本文介绍一个程序，以满足打印表格的需要。

APPLE 虽然没有提供现成的制表所需的横线、竖线以及各类边角的打印方法，但我们可以通过其它途径来实现，如用绘图语言、汇编语言。本程序是利用了打印机的控制码，用控制码中的位像码画出制表所需的 11 个符号，分别写在 11 个子程序中。主程序要用到这些符号时，只要用 GOSUB 语句调用即可。

下面是子程序：

```
10000 PRINT CHR$(27); "K"; CHR$(R);
      CHR$(0):
10010 FOR I = 1 TO R: PRINT CHR$(16);:
      NEXT
10020 RETURN
10100 R(1)=0:R(2)=0:R(3)=0:R(4)=0:R
      (5)=30:R(6)=16:R(7)=16:R(8)=
```

```
        16:R(9)=16:R(10)=16:GOTO 20000
10200   R(1)=16:R(2)=16:R(3)=16:R(4)=
        16:R(5)=30:R(6)=0:R(7)=0:R(8)=
        0:R(9)=0:R(10)=0:GOTO 20000
10300   R(1)=0:R(2)=0:R(3)=0:R(4)=0:R
        (5)=254:R(6)=16:R(7)=16:R(8)=
        16:R(9)=16:R(10)=16:GOTO 20000
10400   R(1)=16:R(2)=16:R(3)=16:R(4)=
        16:(5)=254:R(6)=0:R(7)=0:R(8)=
        0:R(9)=0:R(10)=0:GOTO 20000
10500   R(1)=16:R(2)=16:R(3)=16:R(4)=
        16:R(5)=30:R(6)=16:R(7)=16:R(8)
        =16:R(9)=16:R(10)=16:GOTO 20000
10600   R(1)=0:R(2)=0:R(3)=0:R(4)=0:R
        (5)=240:R(6)=16:R(7)=16:R(8)=
        16:R(9)=16:R(10)=16:GOTO 20000
10700   R(1)=16:R(2)=16:R(3)=16:R(4)=
        16:R(5)=240:R(6)=0:R(7)=0:R(8)
        =0:R(9)=0:R(10)=0:GOTO 20000
10800   R(1)=0:R(2)=0:R(3)=0:R(4)=0:R
        (5)=254:R(6)=0:R(7)=0:R(8)=0:R
        (9)=0:R(10)=0:GOTO 20000
10900   R(1)=16:R(2)=16:R(3)=16:R(4)=
        16:R(5)=240:R(6)=16:R(7)=16:R
        (8)=16:R(9)=16:R(10)=16:GOTO
        20000
11000   R(1)=16:R(2)=16:R(3)=16:R(4)=
        16:R(5)=254:R(6)=16:R(7)=16:R
        (8)=16:R(9)=16:R(10)=16
20000   PRINT CHR$(27); "K"; CHR$(10);
        CHR$(0);
20010   FOR I=1 TO 10: PRINT CHR$(R (I));:
        NEXT
20020   RETURN
```

这 11 个子程序的功能分别为：

SUB 10000 is print —
SUB 10100 is print
SUB 10200 is print ┐
SUB 10300 is print ├
SUB 10400 is print
SUB 10500 is print
SUB 10600 is print └
SUB 10700 is print
SUB 10800 is print |
SUB 10900 is print ⊥
SUB 11000 isprint +

子程序 1000 是打印横线"—"，线的长度可根据需要自行设定，即设定 R 的值。当 R=6 时，相当于一个字符的宽度。

有了这些子程序，在制作表格时就可以画出封闭的框线。下面是一个调用的例子：

```
100   PR # 1: PRINT CHR$(27); "3"; CHR$
      (18);:PRINT
200   GOSUB 10100: R = 18: GOSUB 10000:
      GOSUB 10500: GOSUB 10000: GOSUB
      10200:PRINT
300   GOSUB 10800: PRINT "1";: GOSUB 10800:
      PRINT"2";:GOSUB 10800:PRINT
400   GOSUB 10300: GOSUB 10000: GOSUB
      11000: GOSUB 10000: GOSUB 10400:
      PRINT
500   GOSUB 10800: PRINT "3";: GOSUB 10800:
      PRINT"4";:GOSUB 10800:PRINT
600   GOSUB 10600: GOSUB 10000: GOSUB
      10900: GOSUB 10000: GOSUB 10700:
      PRINT
900   PR# 0: END
```

1	2
3	4

程序的 100 语句不可缺少，这一句的功能是改变打印的行距。若没有这一句，竖线仍为虚线。

本程序用的是 FX—100+打印机，其它型号的打印机打印的结果不能令人满意的话，只要根据控制码略作一些修改即可。

用BASIC语言解决HANOI塔问题

杭州西湖区上泗中学　　林峰

河内（HANOI）塔也叫梵塔，传说它是印度教的主神梵天创造世界时留下的，在一个黄铜板上插着三根细的宝石柱，梵天在其中的一根柱子上从下到上放下了由大到小的六十四片金片。不论白天黑夜，都有一个值班的僧侣按照梵天不渝的法则，把这些金片在三根柱上移来移去：一次只能移一片，并且要求不管在哪一根柱上，小片永远在大片的上面，当所有六十四片都从梵天创造世界时所放的那根柱上移到另外一根柱上时，世界的末日就到了。

我们且不去管这个问题是否与“世界末日”问题有关，只考虑上面的移动金片的过程如何，用BASIC语言如何解决。假设把细柱编号为1、2和3，而柱1上有64个圆盘，从上向下编号为1、2、…、64个，要移到柱2上去（如图，为简单起见，这里只画了九片），我们把这个任务表示为：

movetower(64,1,2)

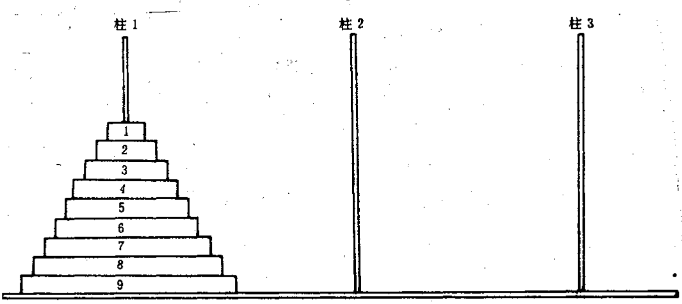


图1　Hanoi 塔

如果将柱1上的顶上63个圆盘移到柱3上，则需将最后一个圆盘从柱1移到柱2，再将63个圆盘从柱3移到柱2上，即得到一个等价过程

movetower(63,1,3)；

把第64片（即最后一片）从柱1移到柱2

movetower(63,3,2)

这三步前后两步与原来问题相似，只是规模小了，而中间一步是一个直接移动的过程，类似的要完成

movetower(n,a,b,c)

其表示“把n个圆盘从细柱a移到细柱b，使用细柱c作为暂时用的塔。”于是可分三步执行：

movetower(n－1,a,c,b)；

把第n个圆盘从a移到b；

movetower(n－1,c,b,a)；

注意当n＝1时是不用这样化的，可直接从a移到b，而移动的盘号为1。

可见这是个递归过程，如果用PASCAL，C和PL/1语言，那是很简单的。很遗憾，BASIC语言不支持递归调用过程，要用BASIC语言解决这个问题可用数组作为堆栈来存放第二步和第三步要完成的内容，而且先解决第一步内容。但如果第一步中只有一个圆盘，就可直接解决，否则第一步还得用类似方法化下去。

这里用数组NUM记录移动的盘片数目，而用NUM(I)＝－1表示可直接移动该盘，要移动的盘号n，用数组SC记录各步的起始柱号，用数组DT记录各步的目标柱号，暂存柱号可以不记录，这样当n>1时，

movetower(n,a,b,c)

可表示为：

NUM(TP)＝n,SC(TP)＝a,DT(TP)＝b

而①movetower(n－1,a,c,b)

②把第n片从a移到b

③movetower(n－1,c,b,a)

可记录为：

①NUM(TP＋2)＝n－1,
SC(TP＋2)＝a,DT(TP＋2)＝c

②NUM(TP＋1)＝－1,SC(TP＋1)＝a,
DP(TP＋1)＝b

③NUM(TP)＝n－1,SC(TP)＝C,DT(TP)＝b

并且把指针TP移到第一步，即TP＝TP＋2。因为a＋b＋c＝6，所以c＝6－a－b，只要记住a和b即可。

当上述n＝1时，就可直接打印移动盘号，移动过程，并使TP⇐TP－1，若NUM(TP)＝－1表示这一步也可直接移动，而移动的是第NUM(TP－1)号圆盘，并使TP⇐TP－1，显然TP是指向当前所要完成的任务，即堆栈的顶。

那么作为堆栈的数组NUM，SC，DT要求多大呢？其实移动n个盘可以化为等价的三步过程，并解决移动n－1个盘的移动问题，所以数组最大要求为2＊(N－1)＋1＝2＊N－1（其中N为总盘数）。

程序中N表示总盘数，S表示刚开始的源柱号，D表示最终的目标柱号，这些量需要键盘输入。PN表示要移动的盘号，S_1为当前源柱号，D_1为当前的目标柱号，在显示/打印中输出。

此程序在Apple-II的Applesoft BASIC下实现，下面

是程序清单及移动 3 个盘从柱 1 到柱 2 的结果。

程序

```
10    INPUT "N=";N:TNPUT"S=";S:INPUT "D=";
      D:LEG=1 +2 * (N -1)
15    IF (N= 0 )OR (S =D) THEN PRINT "MOVING
      NUMBER";COUNT:END
20    DIM NUM(LEG),SC(LEG),DT(LEG)
70    TP =1: NUM (1) =N: SC (1) =S: D (1) =D:
      COUNT =0:GOSUB 100
80    PRINT "NUMBER OF MUVING THE DISCS:";
      COUNT
90    END
100   IF TP <1 THEN RETURN
110   IF NUM(TP) )1 THEN 160
120   IF NUM (TP) =1 THEN PN =1:GOTO 1000
130   PN=NUM(TP-1) + 1:GOTO 1000
160   NUM(TP+2) =NUM (TP) -1: SC(TP+2) =
      SC (TP):DT (TP +2)=6 -DT(TP) -SC(TP)
170   NUM(TP +1) = -1: SC(TP+1) =SC(TP):
      DT(TP +1) =DT(TP)
180   NUM(TP) =NUM(TP +2):SC (TP) =DT(TP
      +2):TP =TP +2: GOTO 100
1000  S1 =SC (TP): D1 =DT (TP): TP=TP -1:
      COUNT =COUNT +1
1010  PRINT,PN,S1;"→";D1
1020  GOTO 100
```

运行结果

```
N=3
S=1
D=2
1        1→2
2        1→3
1        2→3
3        1→2
1        3→1
2        3→2
1        1→2
NUMBER OF MOVING THE DISCS:7
```

如果要在屏幕上形象地显示移动盘片的过程，也可利用 Apple-II 的 BASIC 作图功能加以改进，即可得到一个比较好的演示程序。

在 20 句和 70 句之间插入：

```
30 DIM A(3):DEF FN PP (X) =(X-1) * 1 3+7:A (S)=N
40 GR :COLOR =15:HLIN 0,39 AT 30 : VLIN 29,10 AT 7:
VLIN 29 ,10 AT 20: VLIN 29,10 AT 33
50 COLOR =1:L=N+1:CEN=FN PP(S)
60 FOR I=29 TO 30-2*N STEP -2:L=L-1: FOR J
=0 TO 1: HLIN CEN -1, CEN-L, AT I-J: HLIN CEN
+1,CEN +L AT I-J:NEXT:NEXT
65 RETURN
```

在 1010 句和 1020 句之间插入：

```
1012 L=PN:COLOR=0:CEN=FN PP( S1 ) :YP=30-2 * A(
S1 ):A( S1 )=A( S1 )-1:GOSUB 1030
1014 COLOR=1: CEN =FN PP ( D1 ): YP =28-2 * A (
D1 ):A( D1 )=A( D1 )+1:GOSUB 1030
```

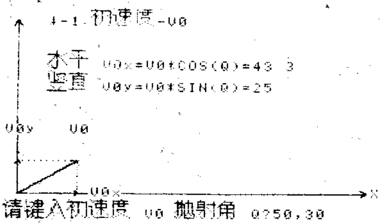
再补上：

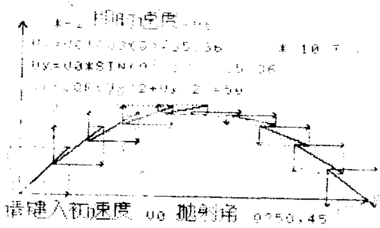
```
1030 FOR I=0 TO 1: HLIN CEN-1, CEN -L AT YP +I:
HLIN CEN+1,CENN +L AT YP +I:NEXT :RETURN
```

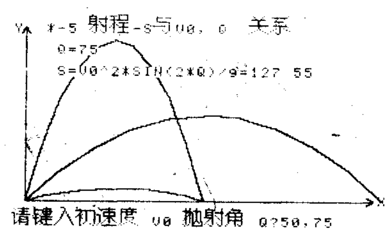
如果演示速度太快，可置一个延时过程，并加上声音，但由于显示器屏幕低分辨率作图下范围的限制，本程序设置为最多演示六个盘的移动过程。

总的移盘次数为 2^N-1，可用级数的知识推导出来，这里不再详细推导。

（上接 37 页）







《抛体运动》辅助教学程序

天津一〇九中　马基

本程序应用STC软汉字系统，用扩展BASIC语言编制。在APPLE－Ⅱ机上DOS3.3操作系统支持下运行。程序包括《平抛》,《斜抛》及相应的自查练习四个功能模块。程序运行时采用两级菜单选择方式。因为斜抛运动规律是曲线运动知识的综合，所以本文以下仅以《斜抛运动》程序说明程序的功能、使用及设计思想。

由主控程序选择《斜抛运动》后，该程序调入内存。屏幕显示二级菜单，选择1后，展示斜抛条件，在汉字提示操作下，逐一地从五个问题角度供使用者讲解，分析斜抛规律。

程序设计原理和特点：

1. 为了精确地从图象和数据两方面研究物体运动时间和位置的定量关系，程序在高分辨率作图时摸拟屏闪摄影的方式。以屏幕平面建立XOY平面座标。根据飞行时间

$$T=\frac{2\cdot V_0 \mathrm{Sin}\theta}{g}$$

等分为T/10为一个时间单位。物体理想化为质点，计算机根据输入的初速度V_O，抛射角θ，计算物体在T/10,T/5,3T/10…T十个时刻的有关力学量。这样屏幕绘图效果既能表现质点运动的过程，又能反映了所研究问题各力学量的定量关系，好象瞬间完成的屏闪照片处理结果。

2. 斜抛运动规律程序通过五个问题来研究。每个问题屏幕上方显示标题，使用者输入V_0，θ的数值，屏幕就显示此问题有关的物理公式和结果以及运动图象。程序以高分辨作图的一个点为长度米的单位，受屏幕绘图范围限制，物体运动显示的范围在水平260m，竖直150m内，超出此范围的质点屏幕不能显示。这样，V_0取值为50(m/s)时，无论θ取什么值($0°\leq\theta\leq 90°$)，图像最佳。

3. 只要不按“0”键，每个标题下程序都可重复执行。按下其它任意一键，程序等待输入新的V_0。θ这样可以满足不同V_0，θ时运动情况的分析比较，重复教学的需要。按入“0”键，进入下一个问题研究。

4. 在即时速度和任一时刻位置两问题中，输入V_0，θ值后，物体在十个时刻的有关力学量及矢量图逐点显示。只要按一次除“0”以外的键，就绘出一时刻的有关图象，并标明对应的时间。按十次，各点逐次完成。这样可以突出物体运动过程中V_t矢量方向，数值变化关系，同时动态表现矢量绘图过程。对于教学难点突破具有很大的帮助。

程序说明：

10～280语句为主程序。其结构安排为，每一个研究的问题设定“旗标”，FL$=“1”，……FL$=“5”。以便在调用公用子程序时作相应的处理。程序中所用汉字先由25～30语句调入内存。在程序中以汉字DATA语句（清单中省略）读入汉字字符串组P$(45)中。子程序包括：500～550打印屏幕位置子程序，1000～1065建立坐标，输入V_0，θ数据程序。1100～1195绘矢量V_t程序。1200～1215矢量绘图中确定某时刻位置嵌套子程序。1300～1380绘物体运动轨迹程序，1400～1445绘质点位置到X,Y轴垂线子程序。

```
10 REM * * * * * * * *
15 REM * PROGRAM-2 *
20 REM * * * * * * * *
25 A$="ZK-DA": PRINT CHR$ (4);"BLOAD";A$
30 HIMEM: PEEK (43635) * 255+PEEK (43634)
35 PRINT CHR$ (26)
40 DIM P$ (45):FOR I=1 TO 45:READ P$(I):NEXT
45 CX=10:CT=140:G=9.8
55 H=10:V=1:PE$=" * "+P$(1)+P$(2)+" * ":GOSUB
   510
60 H=10: V=5: PE$= "1." +P$(1) +P$(2) +P$(38):
   GOSUB 510
65 H=10:V=6:PE$="2."+P$(39):GOSUB 510
70 H=10:V=7:PE$="3."+P$(40): GOSUB 510
75 H=10:V=9:PE$=P$(32):GOSUB 510
80 GET W$:PRINT CHR$ (26)
85 IF W$="3" THEN PRINT CHR$ (4);"RUN MAENU"
90 IF W$="2"THEN PRINT CHR$ (4);"RUN PROGAME-4"
95 H=1:V=2:PE$=P$(1)+P$(2)+P$(6):GOSUB 510
100 H=5:V=4:PE$=P$(7)+P$(27)+"V0 3K"+P$(29):
GOSUB 510

102 H=4: V=5: PE$=P$(8) +P$(9) +P$(10) + "Q":
    GOSUB 510
105 H=10:V=10:PE$=P$(31):GOSUB 510: GET W$
110 H1=5: V1=1: PT$=" *-1"+P$(27)+"-V0": GOSUB
```

```
    1000
115 H1=1; V1=6; P1$="VOy"; H2=9; V2=9; P2$="
    VOx"
117 H3=7;V3=6;P3$="V0";GOSUB 1220; GOSUB 1100
120 H=5; V=3; PE$=P$(29) + "V0x =V0 * COS (Q) =";
    FP$="1";GOSUB 510
122 PRINT INT (100 * V0 * COS (Q1)+.5)/100
125 H=5; V=4; PE$=P$(30) + "VOy =V0 * SIN (Q) =";
    GOSUB 510
127 PRINT INT (V0 * SIN(Q1) * 65+.5)/100;FP$="0"
130 GET W$
135 IF W$<>"0" THEN 110
140 FL$="2"
145 H1=5; V1=1; PT$=" * -2"+P$(28) + "-Vt"; GOSUB
    1000; GOSUB 1300
150 N=0; GOSUB 1200; GOTO 160
155 N=N+1; GOSUB 1200
160 H=3; V=2; PE$="Vx =" + "V0 * COS(Q) =";
    FP$="1"; GOSUB 510
162 PRINT INT (100 * V0 * COS (Q1) +.5)/100;
    " *";N;" T *"
165 H=3; V=3; PE$="VY =" + "V0 * SIN(Q)-g * t
    ="; GOSUB 510
167 PRINT IN8 ((V0 * SIN (Q1) - G * N * E/
    10) * 65+.5)/100
170 H = 3; V = 4; PE$= "Vt =" + "SQR (
    Vx^2 +Vy^2 )="; GOSUB 510
172 PRINT INT (100 * SQR ((V0 * COS (Q1)) ∧
    2+ (V0 * SIN (Q1) - G * N * E/10) ∧ 2+
    .5)/100;FP$="0"
175 GET W2$; IF W2$ <>"0" THEN 155
180 CX=10;CY=140
185 H1=5; V1=1; PT$=" * -3" +(41) +P$(22) +
    P$(23)+P$42); GOSUB 1000
190 GET W2$; IF W2$ <>"0" THEN 155
180 CX=10;CY=140
185 H1=5;V1=1;PT$=" * -3"+P$(41)+P$(22)+
    P$(23)+P$(42); GOSUB 1000
190 FL$="3"; GOSUB 1300
195 H1=4;V1=1;PT$=" * -4"+P$(43)+"-H."+
    P$(44)+"-S."+P$(26)+"-T"
197 GOSUB 1000
200 FL$="4"; GOSUB 1300
205 Y= Vo^ 2 * SIN ( θ1)^ 2/2/G; X= V0^ 2 * SIN
(2 * Q1) /G
210 FOR I=Y TO 0 STEP - 4;SX=X/2 * SC+CX;
    SY=CY-I; IF I=9 * Y/10 THEN HPLOT TO
    SX,SY
215 HPLOT SX,SY; NEXT I
220 H=20;V=7;PE$="H"; GOSUB 530;H=18;V=
    9;PE$="S";GOSUB 530
225 H= 5; V=2; PE$= "H= V0^2 * SIN(Q) ^ 2/ (2
    * g)/g=";FPSKG- * 2|="1"; GOSUB 520
227 PRIMT INT (100 * ( V0^2 * SIN (Q1)∧2 /2/
    G)+.5) / 100; GOSUB 530
230 H=5; V=3; PE$= "S=V0 2 * SIN (2 * Q) /g=";
    GOSUB 520
232 PRINT INT (100 * ( V0^ 2 * SIN (Q1 * 2) /G) +
    .5)/100; GOSUB 530
235 H=5; V=4; PE$= "T=2 * V0 * SIN (Q) /g=";
    GOSUB 520
237 PRINT INT (100 * (V0 * 2 * SIN (Q1) /G+.5) /
    100;FP$="0"
240 GET W4$; IF W4$<>"0" THEN 195
450 PRINT CHR$(26);FL$="5"
250 H1=4;V1=1;PT$=" * -5"+P$(44)+"-S"+
    "3K"+"V0,"+"Q."+"8/"
252 GOSUB 1000
255 FP$="1"
260 H=5; V=3; PE$= "S= V0^ 2 * SIN (2 * Q) /g
    ="; GOSUB 520
262 PRINT INT (100 * ( V0^ 2 * SIN (Q1 * 2) /G) +
    .5)/100; GOSUB 530
265 FV=FV+1
270 GET W5$; IF W5$<>"0" THEN 250
275 PRINT CHR$ (26);FL$="0";FV=0;GOTO 55
280 END
500 REM ? HTAB VTAB,PE $
510 HTAB H ; VTAB V ;PRINT CHR $ (9)
520 IF FP $ ="1"THEN 540
530 HTAB H; VTAB V ;PRINT PE $ ;GOTO 550
540 HTAB H;VTAB V; PRINT PE $ ;
550 RETURN
1000 REM INPUT V0 ,Q
1005 IF FV < >0 THEN 1045
1010 IF FL $ = "5" THEN H =H1; V=V1; PE $ =PT $ ;
     GOSUB 530;GOTO 1020
1015 PRINT CHR $ (26); H=H1; V=V1; PE $ =PT $ ;
     GOSUB 530
1020 H1=12;V1=9;H2=15;V2=1;H3=35;V3=9
1025 H=1;V=1;PE $ ="Y";GOSUB 530
1030 H=1;V=1;PE $ ="Y";GOSUB 530
1035 H=35;V=9;PE $ ="X";GOSUB 530
1040 HPLOT CX, 10 TO CX, CY TO 270 , CY ; HPLOT CX -
```

```
3 ,15 TO CX ,10 TO CX +3,15
1042 HPLOT 265,CY -3 TO 270 ,CY TO 265,CY +3
1045 HTAB 1; VTAB 10; PRINT P$(33); P$(27); "V0";
     P$(3);"Q"
1047 INPUT SV, SD
1050 IF FL $="5" THEN 1060
1055 FL$="2";GOSUB 1300;GOTO 1065
1060 FL$="5";GOSUB 1300
1065 RETURN
1100 REM SHIIAN
1105 IF FT$="1"THEN X =V0 *COS (Q1):Y=V0 * SIN
     (Q1)-9.8 * (N*E/10);GOTO 1115
1110 X=V0 * COS (Q1);Y=V0 * SIN (Q1)
1115 SX=X * SC+CX;SY=CY-Y
1120 IF SX < 0 OR SX >279 OR SY<0 OR SY> 191
THEN 1195
1125 IF IF FL$="3"THEN 1150
1130 HPLOT CX , CY TO SX , SY ; HPLOT CX , CY TO CX,
     SY;HPLOT CX,CY TO SX,CY
1135 IF XT > V0 * COS (Q1) *E /2 THEN 1145
1140 HPLOT CX -2,SY +2 TO CX ,SY TO CX +2,SY +2
1142 HPLOT SX -2, CY -2 TO SX , CY TO SX -2, CY +
     2;GOTO 1150
1145 HPLOT CX -2, SY -2 TO CX , SY TO CX +2, SY -
     2; HPLOT SX -2, CY -2 TO SX , CY TO SX -2, CY
     +2
1150 FOR I=X TO 0 STEP -X/10
1155 SX=I * SC+CX;SY=CY-Y
1160 IF I=9 * X/10 THEN HPLOT TO SX ,SY
1165 HPLOT SX,SY
1170 NEXT
1175 FOR I=Y TO 0 STEP -Y/10 : SX=X * SC +CX :
     SY=CY-I
1180 IF I=9 * Y /10 THEN HPLOT TO SX , SY ; GOTO
     1190
1185 HPLOT SX ,SY
1190 NEXT
1195 RETURN
1200 REM PLOTVt,Vyt,Vxt
1205 FT$="1";XT=(N * E /10) * V0 * COS (Q1)
1207 YT=(N * E /10 ) * V0 * SIN (Q1) -G /2 * (N
     * E/10) * (N * E/10)
1210 CX=XT+10 : CY=140-YT;GOSUB 1100
1215 RETURN
1220 REM ? V,VX,VY
1225 H=H1 : V=V1 : PE$=P1$;GOSUB 530
1230 H=H2 : V=V2 : PE$=P2$;GOSUB 530
1235 H=H3 : V=V3 : PE$=P3$;GOSUB 530
1240 RETURN
1300 REM PLOT (DANDO)
1305 SC=1.05 : FL=0
1310 V0=SV : Q=SD
1315 Q1=Q * 3.14159 / 180 : E=2 * V0 * SIN (Q1) /
G
1320 IF FL $="5"AND COS (Q1)>=.866 OR Q=45 THEN
     TE=E+E/10 : GOTO 1330
1325 TE=E
1330 FOR T=0 TO TE STEP E/10
1335 X=V0 * T * COS (Q1) : Y=V0 * T * SIN (Q1)
     -G /2 * T * T
1340 SX=X * SC+CX : SY=CY-Y
1345 IF SX < 0 OR SX>279 OR SY<0 OR SY> 191
     THEN 1375
1350 IF FL =1 THEN 1360
1355 HPLOT SX ,SY;FL=1
1360 IF FL $="3"THEN GOSUB 1400
1365 IF FL $="5"THEN HPLOT TO SX,SY;GOTO 1375
1370 HPLTO SX ,SY
1375 NEXT T
1380 RETURN
1400 REM X,Y
1405 HPLOT SX,SY
1410 FOR I=X TO 0 STEP-2 : SX=I * SC+CX : SY=CY
     -Y
1412 IF I=9 * X /10 THEN HPLOT TO SX ,SY
1415 HPLOT SX,SY;NEXT I
1420 FOR I=Y TO 0 STEP-4 : SX=X * SC+CX : SY=
     CY-1
1422 IF I=9 * Y /10 THEN HPLOT TO SX ,SY
1425 HPLOT SX,SY;NEXT I
1430 H=4 : V=2 : PE$="X=V0*COS (Q) *t="; FP$=
     "1";GOSUB 510
1432 PRINT INT (100 * V0 * COS (Q1) * T + .5) /
     100;"*T=";INT (100 * T +.5) /100; "*"
1435 H=4 : V=3 : PE$="Y=V0*SIN (Q) * T-.5*g*t
     ∧2=";GOSUB 510
1437 PRINT INT (100 * (V0 * SIN (Q1) * T-.5 * T∧
2 * G) +.5) /100 : FP$="0"
1440 GET W$;IF W$="0"THEN 195
1445 RETURN
```

由于篇幅所限，仅给出三幅运动图象的拷贝

(下转 34 页)

用字符放大打印高分辨率图象的通用程序

山东潍坊一中　刘善平

通常用计算机绘制一幅有趣的图像挂在房间里，的确是别有一番情趣。有时，我们也需要对某些高分辨率图形进行局部的或全部的放大打印。但是，APPLE－Ⅱ只能利用硬拷贝的方法打印图像，它有一些局限性。为此，笔者编制了用字符放大打印高分辨率图像的通用程序

本程序的原理就是先从相应的内存单元取得图像上某个点的颜色，而后根据其颜色的不同，分别用相应的字符代替加以打印。从而得到放大了的图像。但是如何取得某个点的颜色呢？

在APPLE－Ⅱ的低分辨率的绘图中，有一个SCRN函数，可以取得低分辨率图像的某个点的颜色的代码。但是，要取得高分辨率图像的某个点的颜色，则没有具体的函数。但是，高分辨率图像上的一个点对应于内存单元中的一位，一个内存单元中一个字节是8位，这8位中有7位显示在7个点上，最高位用于配合选择这一单元中每一个点的颜色。但对于黑白电视机最高位则显示不出效果。因为我们的打印机不能打印彩色字符，而只能象黑白电视机那样输出黑白字符，至于显示效果则是由显示出的字符的疏密而决定。所以我们只要判断相应内存单元的前7位上，相应一位的值是0还是1，而后利用相应的字符用打印机打印出来就可以了。

本程序打印时可以任意剪辑，正相或反相打印，纵向或横向打印。

程序的结构是：

1～10句是由用户选择相应的图像及打印方式。

20～30句是显示整个图像。

40～230句是程序主体。其中110～160句将取得的十六进制数转化为8位二进制数码，60～90句控制取点的顺序，170～240句进行打印。

500～530句是连接打印机并选择字符的大小及行距的子程序。

在程序中变量N控制高分辨率图像的页码，M和P控制打印方式。当M=0时为正向打印；当M=1时反相打印；当P=1时横向打印；当P=2时纵向打印。由于是字符打印，所以用户可以修改500～530句的子程序，从而随意控制放大或压缩的程序。

注意：由于打印机的型号不同。每行输出字符的个数不同，在剪辑时以每行点的个数不超过打印机每行输出字符的个数为准。

```
1  TEXT:HOME
2  INPUT"LOAD PIC ON DISK? (Y/N):"LL$:IF LL$=
   "N"THEN 5
3  INPUT"PIC' NAME:";NAME$
4  PRINT:PRINT CHR$(4);"BLOAD"+NAME$
5  LOMEM:24576
6  TEXT:HOME
7  INPUT"PAGE:";N:IF N < >1 AND N < >2  THEN 7
8  INPUT"X1,X2,Y1,Y2:";X1,X2,Y1,Y2
9  INPUT"INVERSE(0/1):";M:IF M< >1 AND M < >0
   THEN 9
10  INPUT "PRINT (1/2):"; P: IF P < >1 AND P < >2
    THEN 10
20  IF N=1 THEN HGR2:POKE-16300,0:GOTO 40
30  HGR:POKE-16299,0
40  X1=INT(X1/7):X2=INT(X2/7):DIM A$(192)
45  GOSUB 500
50  IF P=2 THEN FOR D=X1 TO X2: Y=-1: GOTO
    60
55  PRINT"! ";
60  FOR A=8192*N+(P-1) * D TO 8192*N+(P
    -1)*D+80 STEP 40
70  FOR B=A TO A+896 STEP 128
80  FOR C=B TO B+7168 STEP 1024
85  IF P=2 THEN X =PEEK(C)
90  IF P=1 THEN FOR D=X1+C TO X2+C:X=PEEK
    (D)
100 A$=""
110  FOR E=7 TO 1 STEP-1
120 F=X-INT (X/2)*2
130  IF ABS (F-M) =0 THEN A$ =A$ + "": GOTO
     150
140  A$ =A$ +"M"
150  X=INT(X/2)
160  NEXT E
```

（下转23页）

用微机给知识竞赛计时计分

河北邯郸一中　李书岑

面临着世界上新技术的挑战，我们广大青少年为四化而奋发学习的热潮方兴未艾。各地举办的各种类型的知识竞赛应运而生。这些竞赛都需要计时、计分。间的控制和分数的显示必然需要一定的设备，制做这些设备就要花费一定的人力和财力。鉴于目前微机已较普及，应用微机来承担这项工作既经济又方便，应该说是一种较理想的方法。为此，笔者编制了一个可同完成计时、计分的程序。在APPLE上经反复调试，多次使用，都取得满意的效果。现提供出来，供使用者参考。

本程序对参赛者可在1—6人(组)间任意选择，用低分辨率绘图显示组号、分数和走时，同时用声响和文字窗对走时做进一步地显示。具有显示清晰，计时准确，使用方便等优点。

使用说明：

键入RUN↓后显示“JI BEN FEN=？”，向用户询问各组予先给定的基本分。可键入三位数字的分数值。如键入200↓。

显示“REN SHR=”，询问参赛的人(组)数。可在1—6之间任选。

显示屏幕用低分辨率绘图显示出各组的编号和基本分的值，同时在绘图区的下方绘出一条亮线。

在文字窗显示“SHI JIAN=”，询问抢答或答题规定用的时间值。可键入0—9之间的任意数。根据键入数字的不同，可分下列三种情况：

1. 若是键入1—8之间的数字，则把该数乘以10视为规定的时间秒数。一旦按过此范围的某个值，计时即从此开始。每经过一秒，文字窗显示出时间的数字并发出一声“嘀”的响声，同时绘图区下边的亮线的右边增加一个黑点，累计10秒后，右边的10个黑点消失，而左边出现一个黑点……。若需中途停止走时，只需按下“T”键即可。若在走时过程中没有按过“T”键，就按规定的时间走完为止。计算机将发出一连串的声响表示时间结束，同时绘图区下部的亮线恢复。不论时间中途停止还是走完为止，程序将再次转到询问时间状态。

2. 若键入“0”，则显示“HAO SHU”，询问答题者的号数。当键入号数后又显示“Y/N/0”，询问答题是否正确。正确应键入“Y”给该号加10分，不正确应键入“N”给该号减10分。若分数不加不减，则应键入“0”。计算机将把分数进行修改，之后又回到询问状态。若在输入组号时就键入了“0”，则不显示“Y/N/0”而直接回到询问时间状态。即“0”是询问键入时间和询问号数的切变信号。

3. 若键入“9”，则清除屏幕后结束运行。

本程序除开始询问基本分、人数时用INPUT语句(键入数值后需打回车键)外，以后各项询问时间、组号值时不需再打回车键。这就给使用者带来很大的方便。由于GET后面询问的是数字，所以采用了数值变量，这样一旦键入了数字以外的其它健，将显示出错信息而停机，所以程序采用了ON ERR GOTO　语句加以处理。这样一来，不论错按了什么健，计时计分都能正常进行而不至中途停机。

程序中810语句为延时而设，改变TO后面的数值，可把走时调准。

如果每题得分和扣分要随意确定，可加一条语句用来询问，该语句应为

465 PRINT“FN SHU=”;GET F1:F1=F1x * 10 这样就将键入的数值乘以10做为分数的加减量。

程序：

```
10    DTAT 1,1,2,1,3,1,4,1,4,2,4,3,4,4,4,5,4,6,4,7,
        3,7,2,7,1,7,1,6,1,5,1,4,1,3,1,2,0,0,0,0
20    DATA 4,1,4,2,4,3,4,4,4,5,4,6,4,7,0,0,0,0,0,0,
        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
30    DATA 1,1,2,1,3,1,4,1,4,2,4,3,4,4,3,4,2,4,1,4,
        1,5,1,6,1,7,2,7,3,7,4,7,0,0,0,0,0,0,0,0
40    DATA 1,1,2,1,3,1,4,1,4,2,4,3,4,4,3,4,2,4,1,4,
        4,5,4,6,4,7,3,7,2,7,1,7,0,0,0,0,0,0,0,0
50    DATA 1,1,1,2,1,3,1,4,2,4,3,4,4,1,4,2,4,3,4,4,
        4,5,4,6,4,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0
60    DATA 4,1,3,1,2,1,1,1,1,2,1,3,1,4,2,4,3,4,4,4,
        4,5,4,6,4,7,3,7,2,7,1,7,0,0,0,0,0,0,0,0
70    DATA 4,1,3,1,2,1,1,1,1,2,1,3,1,4,2,4,3,4,4,4,
        4,5,4,6,4,7,3,7,2,7,1,7,1,6,1,5,0,0,0,0
80    DATA 1,1,2,1,3,1,4,1,4,2,4,3,4,4,4,5,4,6,4,7,
        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
90    DATA 1,1,2,1,3,1,4,1,4,2,4,3,4,4,4,5,4,6,4,7,
        3,7,2,7,1,7,1,6,1,5,1,4,1,3,1,2,2,4,3,4,
```

```
100   DATA 3,4,2,4,1,4,1,3,1,2,1,1,2,1,3,1,4,1,4,
      2,4,3,4,4,4,5,4,6,4,7,3,7,2,7,1,7,0,0,0,0
110   DATA 0,5,10,24,29,34,0,5,10,24,29,34,0,5,10,
      24,29,34
120   DATA 0,0,0,0,0,0,13,13,13,13,13,13,26,26,26,
      26,26,26
130   DATA 1,10,25,10,27,10,1,23,3,23,5,23,25,23,
      27,23,29,23,31,23,1,36,3,36,5,36,7,36,9,36,
      25,36,27,36,29,36,31,36,33,36,35,36
140   HOME
150   INPUT"JI BEN FEN=? ";E
160   INPUT "IEN SHU =";R
170   HOME
180   ONERR GOTO 400
190   F1=10
200   DIM H(9,20),P(9,20),N(18),X(18),Y(18),A(18)
      B(18),W(18)
210   FOR K =0 TO 9
220   FOR L =1 TO 20
230   READ H(K,L),P(K,L)
240   NEXT L:NEXT K
250   GR : COLOR =7
260   FOR K = 1 TO 18: READ X(K): NEXT
270   FOR K = 1 TO 18 :READ Y(K): NEXT
280   FOR K = 1 TO R :A (K)= E:B(K)=E:NEXT
290   FOR K =1 TO R
300   FOR L =1 TO K
310   READ D1,D2 : PLOT D1,D2
320   NEXT L: NEXT K
330   N1=1:N2 =R * 3
340   FOR K =1 TO R: ZP =100
350   FOR L =1 TO 3
360   PQ =PQ +1
370   N(PQ)=INT(B(K)/ZP):B(K)=B(K)-N(PQ) * ZP:
      ZP=ZP/10:W(PQ)-N(PQ)
380   NEXT L:NEXT K: GOSUB 550
390   GOSUB 700
400   PRINT : PRINT "HAO SHU= (1---"; R; ")";:
      GET F:PRINT F
410   IF F < > INT (F) OR F <0 OR F >R THEN 400
420   IF F =0 GOTO 530
430   PRINT "(Y/N/0)";:GET S$: PRINT
440   IF S$ ="Y"OR S$="N" OR S$="0" THEN 460
450   GOTO 430
460   IF S$="0"THEN 530
470   IF S$ = "Y" THEN A (F) =A (F) +F1: B (F) =A
      (F):GOTO 490
480   A(F) =A(F)-F1:B(F)=A(F)
490   N1 = (F -1) * 3 +1:N2 =N1 +2
500   ZP =100: FOR K =N1 TO N2
510   N(K) =INT (B(F) /ZP ):B(F) =B(F)-N(K) *
      ZP:ZP =ZP/10
520   NEXT K:GOSUB 550
530   GOSUB 700
540   GOTO 400
550   FOR Z =N1 TO N2
560   IF W =0 THEN 630
570   IF N(Z) =W(Z) THEN GOTO 680
580   COLOR =0
590   FOR K =1 TO 20
600   IF H(W(Z),K) =0 THEN 630
610   PLOT H(W(Z),K) +X(Z),P(W(Z),K) + Y(Z)
620   NEXT K
630   COLOR=7
640   FOR K =1 TO 20
650   IF H(N(Z),K) =0 THEN 680
660   PLOT H(N(Z),K) +X(Z),P(N(Z),K) +Y(Z):W
      (Z) =N(Z)
670   NEXT K
680   NEXT Z
690   W =1: RETURN
700   S =1:S1 =20
710   COLOR =7: HLIN 0,39 AT 38: HLIN 0,39 AT 39
720   PTINT "SHI JIAN =";: GET AB: PTINT
730   IF AB =0 THEN 840
740   IF AB =9 THEN TEXT :HOME :END
750   FOR K =0 TO AB * 10
760   IF PEEK (-16384) =212 THEN GET K$: GOTO
      700
770   PRINT K: IF K =0 THEN 830
780   COLOR =0:VLIN 38,39 AT S1:S1=S1 +2
790   IF S1 >39 THEN S1 =20: VLIN 38, 39 AT S: S =S
      +2: COLOR =7: HLIN 20, 39 AT 38: HLIN 20, 39
      AT 39
800   PRINT CHR $ (7)
810   FOR SD =1 TO 450: NEXT SD
820   IF K =AB * 10 THEN FOR L=1 TO 10: PRINT CHR
      $ (7): NEXT
830   NEXT K: PRINT K -1 :GOTO 700
840   RETURN
```

介绍一种新型的微机 I/O 接口芯片

王应东

SY6532 数据块与 SY6500 微处理机系列一起使用。它由 128×8 字节的 RAM、两个由软件控制的 8 位双向数据口(它允许在微处理机和外设间直接地进行信息交换)、一个可编程的具有定时中断能力的时间定时器、时钟周期和边缘检测中断电路所组成。

一. 主要功能如下:

- 用于直接同微处理机通讯的 8 位双向数据总线。
- 可编程的边缘检测中断
- 128×8 字节的 RAM
- 用于同外设接口的两个 8 位双向数据口
- 两个可编程的 I/O 数据方向寄存器
- 可编程的时间定时器
- 可编程的定时中断
- TTL 和 CMOS 兼容的外部连接线路
- 具有直接地驱动晶体管的输出脚
- 具有高阻态的三态数据引脚

二. 内部结构框图及电气参数说明(表一、二)

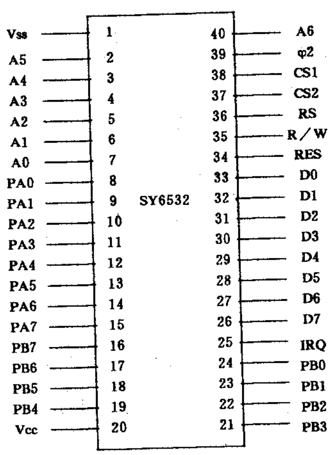


图1 管脚排列图

表一、最大额定值

额　定　值	符号	电　压	单位
输入电压	V_{CC}	−0.3—+7.0	V
I/O 电压	V_{IN}	−0.3—+7.0	V
操作温度范围	TOP	0—+70	C°
存贮器温度范围	TSTG	−55—+150	C°

表二、　电气特性

特　性	符号	最小值	平均值	最大值	单位
输入高电平	VIH	2.4		VCC	V
输入低电平	VIL	−0.3		0.4	V
输入漏电流 $\left\{ VIN=+5V;\ A_0\sim A_6, \overline{RS}, R/W, \overline{RES}, \phi2, CS_1, \overline{CS_2} \right\}$	IIN		1	2.5	uA
高阻时输入电流 $\left\{ VIN=0.4V—2.4V;\ D_0\sim D_7 \right\}$	ITSI		±1.0	±10.0	uA
输入高电流 $\left\{ VIN=2.4V;\ PA_0—PA_7, PB_0—PB_7 \right\}$	IIH	−100	−300		uA
输入低电流 (VIN=0.4V) ($PA_0—PA_7, PB_0—PB_7$)	IIL		−1.0	−1.6	mA
输出高电压:(VCC=最小值,ILOAD ≤−100uA) ($PA_0—PA_7, PB_0—PB_7, D_0—D_7$) (ILOAD≤3mA ($PA_0—PA_7$)	VOH	;1.5			V
输出低电压 (VCC=最小值, ILOAD≤1.6mA)	VOL			0.4	V
输出高电流 $VOH\geq 2.4V$,($PA_0—PA_7, PB_0—PB_7, D_0—D_7$) / $V_{OH}\geq 1.5V$,(可直接驱动三极管,$PB_0—PB_7$)	IOH	−100 / +3.0	−1000 / +5.0		uA / mA
输出低电流 $V_{OL}\leq 0.4V$	IOL	1.6			mA

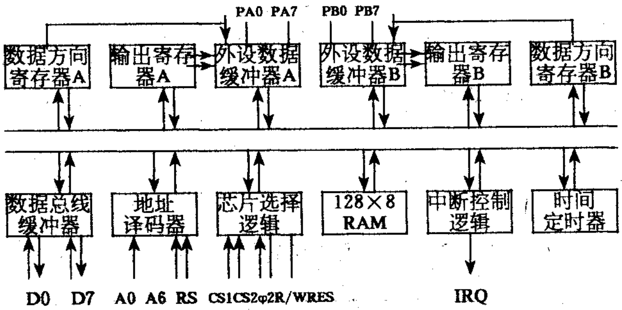


图 2　内部结构框图

三、接口信号的描述

Reset（$\overline{RES}$）：

在系统初始化期间，$\overline{RES}$ 上的逻辑"0"输入将引起 4 个 I/O 寄存器都处于"0"状态。在软件的控制之下，当系统处于一个特定的状态时，它将定义所有的 I/O 总线为输入。在 RESET 期间，数据缓冲区被置为 OFF—STATE 状态，中断被屏蔽，当需要 REST 信号时，$\overline{RES}$ 保持至少为一个时钟周期。

Input clock　：

系统的输入时钟信号。它可以是低电位时钟（$V_{IL}<0.4V, V_{IH}>2.4V$），也可以是高电位时钟（$V_{IL}<0.2V, V_{IH}=V_{CC}{+0.3V \atop -0.2V}$）。

READ/WRITE　R/W：

该信号由微处理机提供。用以传送数据到微处理器和 6532，或从微处理器和 6532 发出数据。R/W 为"高"允许微处理器读（和适当的地址一起）由 6532 提供的数据，R/W 为"低"时允许微处理器写（和适当的地址一起）到 6532。

Interrupt Request　（$\overline{IRQ}$）：

$\overline{IRQ}$ 是来自中断控制逻辑的中断信号，正常情况下为高电平。当 SY6532 发出中断申请时为低电平。$\overline{IRQ}$ 是集开路输出，它允许几个单元"线或"到公用的 $\overline{IRQ}$ 微处理器的输入脚。通过 PA_7 或定时器时间到的跳变可以使 $\overline{IRQ}$ 有效。

Data Bus（D_0—D_7）：

该数据块有 8 条双向数据针，它们接到系统的数据线，用于与微处理器交换数据。除在"读"操作发生的时候，输出缓冲器保持截止状态。

外设数据缓冲器：

该数据块有 16 条用于 I/O 操作的输出针，每一针都可独立地被编程为输入或输出。这 16 针被分为两个 8 位口，PA_0—PA_7，PB_0—PB_7。将"0"写到数据方向寄存器相应的位，可将相应的输出针定义为输入。写"1"，则可将相应的输出针定义为输出。

地址输出针，（A_0—A_6）：

7 个地址输出针。它们和 $\overline{RS}$（寄存器选择），CS_1和$\overline{CS_2}$（片选信号）一起使用，用于选择芯片和寄存器。

RAM：

6532 包含有 128×8 的 RAM。通过 A_0—A_6，$\overline{RS}$$CS_1$，和 CS_2 来寻址。寻址方法见寻址译码表。

寄存器：

该数据块有 4 个内部寄存器：两个数据方向寄存器和两个输出寄存器。数据方向寄存器（A 口和 B 口）控制数据 I/O 的方向。当数据方向寄存器（DDRA 和 DDRB）中某位为"0"，则定义相应 I/O 口的引出针为输入，反之为输出。

时间定时器：

时间定时器有三个基本部份：除法寄存器，可编程的 8 位寄存器和中断逻辑。当定时器被编程为计算时，可达到 256 种定时间隔。每种定时间隔可以是：1T，8T，64T 或 1024T 的增量。（此处的 T 是系统时钟

操作	$\overline{RS}$	R/W	A_4	A_3	A_2	A_1	A_0
写 RAM	0	0	—	—	—	—	—
读 RAM	0	1	—	—	—	—	—
写 DDRA	1	0	—	—	0	0	1
读 DDRA	1	1	—	—	0	0	1
写 DDRB	1	0	—	—	0	1	1
读 DDRB	1	1	—	—	0	1	1
写输出寄存器 A	1	0	—	—	0	0	0
读输出寄存器 A	1	1	—	—	0	0	0
写输出寄存器 B	1	0	—	—	0	1	0
读输出寄存器 B	1	1	—	—	0	1	0
写定时器			—	—			
÷1T	1	0	1	(a)	1	0	0
÷8T	1	0	1	(a)	1	0	1
÷64T	1	0	1	(a)	1	1	0
÷1024T	1	0	1	(a)	1	1	1
读定时器	1	1	1	—	1	—	0
读中断标志	1	1	1	—	1	—	1
边沿检测控制	1	0	0	0	1	(b)	(c)

（下转 52 页）

CEC-I 中华学习机简介

潍坊计算机公司　王进吉

CEC-I 中华学习机是由电子工业部计算机与信息局组织，由清华大学主持联合设计，由国营七三四厂、深圳华明计算机有限公司，陕西省计算机厂等单位参加研制成功的一种微型计算机。该机的特点是结构灵活，其主机配上家用彩色或黑白电视机和盒式录音机，就可以组成基本系统；若插上汉字系统部件和软盘驱动器接口部件（在主机大板上均留有相应位置），再接上软盘驱动器，就可以构成一台功能强，有汉字系统支持且价格低廉的微机系统。

CEC-I 中华学习机的主要性能和技术指标：

一、硬件

1、CPU 6502，时钟频率 1.023MHz，单一+5v 电源供电。数据总线 8 位，地址总线 16 位，直接寻址范围 64KB。

2、RAM 64KB，由两片 64K×4 位的集成电路 HM50464 构成；ROM32KB，包括监控程序，BASIC 解释程序，汉字处理程序等。

3、专用芯片

MMU：存储器管理部件，其主要功能是根据需要选中相应的存储器

IOU：输入输出管理部件，其主要功能是根据显示方式的不同选取相应的显示缓冲区。

PAL 电路：为一可编程的逻辑阵列电路，其主要功能提供主机大板上所需各种时序信号

4、显示方式

中华学习机共有五种显示方式：

(1)文本方式

字符为 5×7 点阵，满屏可显示 24 行×40 列共 960 个字符，有正常、反相、闪烁三种状态。

(2)低分辨率图形方式

可显示 48 行×40 列共 1920 个彩色方块，每个方块可在 16 种不同的颜色中选择

(3)高分辨率图形方式

可显示 192 行×280 列个彩色点，每个点的颜色可在 6 种颜色中选择

(4)混合显示方式：

可以是低分辨率图形方式与文本方式的混合，即 40 行×40 列彩色方块与 4 行×40 列字符的混合显示；也可以是高分辨率图形方式与文本方式的混合，即 160 行×280 列个彩色点与 4 行×40 列字符的混合显示。

各种方式均可通过控制软开关实现转换。

5)汉字显示方式：

每个汉字为 16×16 点阵，满屏可显示 10 行×17 列共 170 个汉字，外加一个状态行；

ASCII 字符为 8×16 点阵，满屏可显示 10 行×34 列共 340 个 ASCII 字符，外加一个状态行。

5. 键盘

中华学习机键盘为打字机式标准键盘，外加一些特殊功能键，共有 69 个键，通过一条 26 线扁平电缆与主机相连是最基本的输入设备。

6. 汉字系统

汉字系统由汉字管理程序与汉字库(包括国际一、二级仿宋体汉字)构成，已提供拼音、区位两种汉字输入方式，另外留有相应的硬件与软件接口供用户自行扩充时应用。

汉字系统可根据用户需要取舍。

7、扩充插槽

主机提供一个与 APPLE 机兼容的扩充插槽，由于汉字系统已占用 3 号插槽，软盘驱动器接口已占用 6 号插槽，因而保留的这个插槽可通过跳接插座定义为 1、2、4、5、7 号，出厂时该插槽定义为 1 号。

8、驱动器接口

在主机板上相应位置插上驱动器接口器件，就可连接一台软盘驱动器。要求配用 $5\frac{1}{4}$ 吋软盘驱动器，使用单面单密度 $5\frac{1}{4}$ 吋软盘，格式化后盘的容量约 140K。

该接口也可视用户需要取舍。

9、录音机接口

主机板已提供一个 5 芯标准 DIN 插座，与录音机连接起来后，无论在监控状态下或 BASIC 状态下都可以很方便地将信息存到磁带上或从磁带上读到内存进行处理。

该接口输入输出特性参数如下：

输入电压：峰值 1V　输入阻抗：12KΩ　(下转 45 页)

浅论 APPLE 机在生油岩热解分析方面的应用

丛仁滋

一、引言

"Rock-eval"Ⅱ、Ⅲ型仪是我国几个大型油田所属的地质研究部门广泛使用的一种用于估算地下储油量的仪器。这种仪器对被分析的岩芯进行升温加热，收集岩芯中的有关生油方面的有机化学成份，并转换为电信号进行处理分析，打印出相应的数据。但是由于仪器所给出的只是分析结果，还不能满足地质研究人员工作中的需求，于是，我们用苹果机Ⅱ机与"Rock-eval"Ⅱ型仪联机开发，利用计算机对生油岩热解分析过程进行定时的数据采集并处理，而且没有破坏仪器原来使用方法的完整性，使用户感到满意。

二、系统的硬件配备

除了苹果机的基本系统以外，还配备了用于作中断源的苹果机专用时钟卡一块，12 位的 A/D 卡一块。

在系统配件的选型上，我们考虑到"Rock-eval"Ⅱ型仪输出至记录仪的模拟信号的最大幅度值为 1V，而用户又要求计算机要能分辨 0.5mv 以上的电信号。如要使用 8 位的 A/D 板，则计算机对于输入的模拟信号的最小分辨能力为：5/225＝19.6mv。如果将输入的模拟信号放大到 8 位 A/D 板能分辨的信号的范围之内，就要将输入信号放大 19.6/0.5＝39.2≃40 倍，而"Rock-eval"Ⅱ型仪器输入的模似信号的最大幅值 1V，经过 40 倍放大就会大于 A/D 板的输入极限值 5V，因此选用 8 位 A/D 板是不能满足要求的。如要选用 12 位 A/D 卡，则计算机所能分辨的最小模拟信号为 5/2048＝2.44mv，再加一级前置放大，以输入信号放大 5 倍，那么 A/D 板可分辨的输入信号最小幅值为 2.44mv/5≐0.49mv＜0.5mv，就能够满足用户的要求，并且仪器输出的模似信号的最大幅值 1V，经过 5 倍放大以后，也仍在 A/D 板允许的输入范围之内。所以我们选用了带有四路前置放大器的 12 位 A/D 板。根据仪器输出的特点，我们用一路 A/D 转换通路作为数据采集通路，而用另一路转换通路，作为数据采集过程的启动信号。由于在数据采集过程中要使用中断服务子程序来完成数据采集工作，我们选用时钟板作为系统可屏蔽中断的中断源来控制两次中断的间隔时间。

由于"Rock-eval"Ⅱ型仪的模拟信号输出幅度一般较大（1mv～1v），而且 A/D 板的前置放大器的放大倍数并不很大，信号干扰也很小，所以在硬件的配备上，我们没有加低通滤波器。

三、软件设计

生油岩热解分析的过程是这样的：首先是三分钟清理管道的时间，其次是在 300℃的条件下保持恒温三分钟，再次是五分钟内坩埚从 300℃～500℃的升温过程，最后是在 550℃的条件下保持恒温一分钟。由于在升温过程中，仪器对放在其内部加温炉中的坩锅是以每 6 秒升温 5℃的速率进行加温的。并且在对坩锅加温的同时，将有关生油的有机化学成分转换成的与之对应的电信号送至记录仪，所以我们在设计数据采集及分析过程的主控程序的结构时采用了这样一个结构，即用机器语言程序完成数据采集工作；用 APPLE SOFT BASIC 语言程序进行数据的计算、处理和数据存盘以及图象显示等工作。由于用户认为在他们研究工作中，坩锅每升温 5℃（其时间间隔为 6 秒钟）计算机采集一个数据便可满足他们对于关于岩芯样品的生油方面的数据要求。于是我们采用了定时使计算机产生一次中断的方法，利用中断服务程序来完成数据采集工作。对于两次中断之间的时间间隔的长短，由苹果机的专用时钟卡提供的可屏蔽中断源提出的中断请求来控制。由于岩芯样品分析过程的特点，要求控制程序对于数据采集的全过程进行图象跟踪显示，并对过程中的每一个分段（300℃条件下的三分钟恒温、300℃～500℃的升温段及 550℃条件下的恒温）进行图形面积积分，每隔 5℃保留升温段中的一个数据并求出这些数据所对应的输入曲线到达最大值时所对应的温度值 Tmax 等等。根据如上这些要求，我们设计了程序的总体结构。为了提高 A/D 板所采集的数据的可靠性，采用了程序滤波，即每一个数据采集点连续采集六次用公式 $A=(\sum_{i=1}^{6}ai-a_{min}-a_{max})/4$，则 A 为本次数据采集点的采样值。

在图形显示上，采用了高分辨全屏幕图形显示，在屏幕拷贝上，使用了倍宽拷贝。对于采集的数据所

进行数值积分计算上，采用了常用的梯形积分。由于起峰区间的图形近似于二次曲线，所以，要采用回归方程的形式时，我们采用了二次方程的形式：

$$Y = aX^2 + bX + C$$

对采集到的起峰区间的数据进行二次回归，并求出热解峰温 Tmax 来。

为了资料管理上的方便，在这个系统中，设计了用于打印报告表、数字表的程序块以及各种样品的曲线重叠程序块，这主要是用于不同样品之间的生油情况的比较和为原始的生油量 S_2 的估计做准备工作。

四、总结

将以上的程序块配接成一个系统，进行了一段时间的运行，其结果说明 APPLE 机和 Rock-eval 联机能起到一个专用微型机的作用，并具有图象跟踪显示、报表打印和数据归档存盘等一些专用微型机所不具有的优点，而且还为以后计算地下岩芯最原始的含油量 S_{212} 和更精确地估算地下储油量作了一些必要的准备工作，从而推广了苹果机的应用范围。

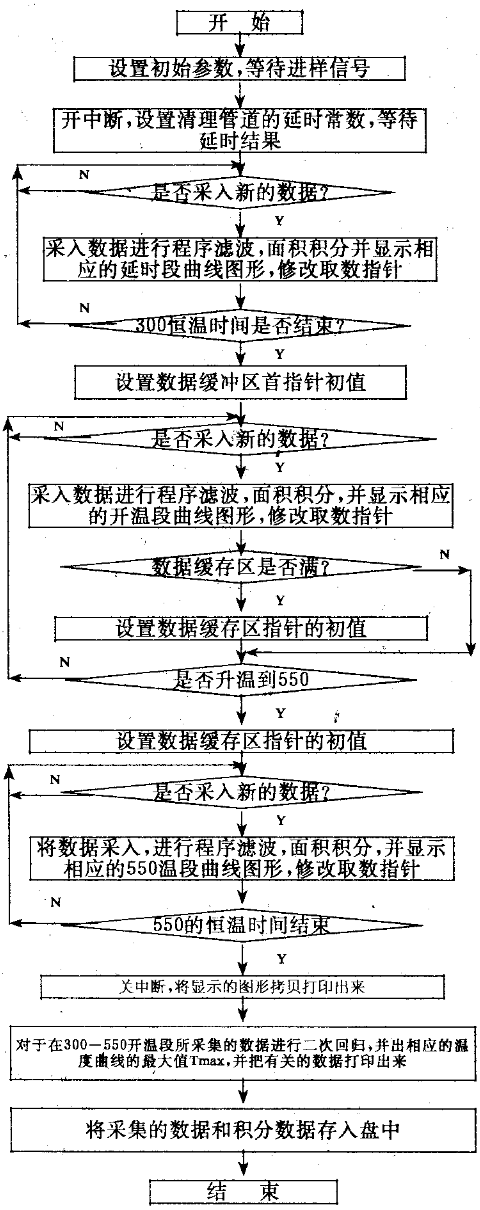


图2 主控制程序框图

（上接 43 页）

输出电压：25mv 输出阻抗：100Ω

10、扬声器接口

主机板上提供的扬声器接口已连接一个 0.25w，8Ω的扬声器。

1 提供三个电平开关信号输入及四个模拟量输入，可以运行原 APPLE 机的绝大多数游戏软件。

二、软件

1、系统软件

主机板上已固化好的系统软件包括监控、BASIC、LOGO（子集）和汉字处理软件。另外，原来 APPLE 机的系统软件如 DOS3.3 磁盘操作系统、CP/M2.2 操作系统、U.C.S.D—P 系统均可在中华学习机上应用。

2、应用软件

除了原 APPLE 机的应用软件外，中华学习机还有一些应用软件和游戏软件，现在可提供的有华容道、小熊学英文、高中物理、音乐黑板、英文打字练习、小学数学、汽车设计师、天文观测台、化学实验室、美术设计师等十余种，以后随着该机的普及，将会有更多富有特色的软件出现。

中华学习机目前全国已有十多家计算机厂家生产，潍坊计算机公司微电脑厂是第一批生产厂家之一，现已批量生产，欢迎全国各地新老用户使用，愿它成为您学习、工作的好助手！

第九章　进一步的故障检测技术

在以上各章中我们已经介绍了许多有关苹果机故障检查的基本技术，可以看出对于苹果机故障的识别和校正有11个最佳步骤；同时，我们也介绍了怎样识别计算机各种元器件的知识，以及寻找故障的四条途径：

1. 硬件的途径；
2. 软件的途径；
3. 自身诊断；
4. Apple 专用方法。

在硬件途径中，使用检查工具，如逻辑探测器和逻辑脉冲发生器逐步检查线路，这就要求我们有测试设备以及某些数字电路的知识。

软件的途径在苹果机修理技术中是一种广泛使用的寻找故障的方法，只要磁盘驱动器能正确地进行自举，使用诊断软件查找芯片的故障是很有效的。

Apple 专用方法使用第四章和第七章的寻找故障指南可以快速地正确指出芯片可能出现的故障，如果断定问题不是出在芯片则还应进一步找出出现故障部分的位置，这时就需应用本章所介绍的技术，测试可疑的故障区域的其余元器件。

在本章中将进一步介绍检查故障的技术，以及维修使用的各种工具，像本书其他部分一样，这章所论述的也是为保持苹果机系统高质量操作状态的实质性资料。

第一节　检测工具与设备

当问题不能只藉助於操作程序与画纸得到解决时，则要进一步藉助於"工具"，这些维修工具不仅只是螺丝刀、偏嘴钳和烙铁头，同时也包括电气测试设备——各种测量仪表（VOM，DVM，DMM），逻辑探针器，逻辑脉冲发生器，电流故障检寻器，接线夹子，示波器以及逻辑和图象分析器。

近年来电气测量设备已有很大的改进，测试和寻找电路故障的效率有了明显的提高。二十年前出现的称为 VOM 的测量仪表（电压—欧姆—毫安表）可以用来测量电器电路的三个参数，即电压、电阻和电流。然后是 VTVM 真空管电压表）。在电气电路让位于电子电路不久，数字电路代替了模拟电路，运用这些新的电路设计出了寻找故障的新型测量仪表。DVM（数字电压表）和 DMM（数字万用表）很快成为数字技术的优先选用测量设备，因为它们提供了适于电子电路测试的很多优良性能，其中也包括测量精度的提高。这种仪表具有高的输入阻抗（或电阻），因此它们不会降低数字电路被测点的电压，也不会增加被测点的负载。

两种变化影响了检查和修理工具的型式。第一，真空管被固体器件如半导体三极管和集成电路（IC），或芯片代替；第二，电路本身是由很多元件紧凑地集成在很小的芯片上，从而它们的体积变得更小了。只需拿早期的收音机或电视机（直立有4呎高，重40磅）与现代的手表式收音机或手表式电视机比较一下，就可以认识到电子线路是更小、更复杂、更难测试。

电子学的进步给电子技术的发展提供了广阔的前景，设计者们立即使用这种技术和新型的元器件制作出了各种灵巧的测试设备。在数字电路的测试中不用担心过多地增加电路负载而引起的不精确读数。也不会由于体积粗大的测试笔在芯片管脚或密集的印刷线路之间造成短路而毁坏电路。

一. 逻辑接线板

逻辑接线板是一种数字电路测试设备。这种方便的工具能在其上面安装集成电路器件，并将其管脚露出在板的上面，测量或监测笔或者微型接线夹子可以接触这些管脚，以便确定所测元件的任意管脚的逻辑电平。

另一种型式的逻辑接线板具有内部监视的性能，除了露出的管脚外，在该接线板的顶部连接了一行发光的二极管，它们不断地显示芯片上每个管脚逻辑状态，这些发光二极管（*LEDs*）由被测电路的电源导通（表示逻辑"1"）。全部管脚上都接有缓冲电路，因而不会增加被测电路的负载。

注意：当使用逻辑接线板时，先关闭电源，装上芯片后，再打开电源（这样可以防止芯片的偶尔短路）。

逻辑接线板可以做成各种不同型式。几乎可适用于全部逻辑元件系列，包括 TTL 和 CMOS 芯片，而其直流电源电压可以达到 30V。

在使用接线板时，分别将顶部的 LED 接线端子插

入管脚接触点，并将接线夹子在被测芯片顶部滑动，当电路通电时，*LEDs* 将指示芯片上的每个管脚的逻辑电平。

逻辑接线板可以用於多达 16 个管脚的集成电路芯片（*ICs*），或者说它适用于 *AppleIIPlus* 母板 80％的集成电路芯片。

二．逻辑探针

当需要确切地研究线路情况时，可以使用逻辑探针。毁坏的芯片是不能修理的。但是逻辑探针可以找出已经损坏的芯片，以便更换。

逻辑探针在电路板的分析中，它是一种使用非常广泛的工具。虽然它不能像复杂的测试设备逻辑分析器那样做那么多工作，但是当电子线路频繁出现的芯片故障，在通电电路中使用这种探针可以简单、可靠和迅速的查找故障。使用这种工具在电路分析中 90％的问题可以得到解决。

当探针的细小尖端与某个被怀疑的芯片管脚接触，或者在电路板上追踪检查某个测试点时，在靠近探针处的指示灯指示出该点的逻辑状态（电平），市售大多数逻辑探测器的金属探针在偶尔接触到高于选通电压（＋5 伏）的电源电压时，有防护功能，不至损坏（可防护高达 120V 交流电压，时间 30 秒）。

某些探测器在它们的探针附近装有两个指示灯，其一是表示逻辑高电平；另一表示逻辑低电平。更好的探测器可以指示哪个测试点出现了脉冲信号，它们还能存入一个短的脉冲串，以指示测试点出现的假信号或是脉冲尖峰。购置逻辑探针，应该选择适合所要分析的逻辑芯片系列的那一种型号。

对於电路的诊断分析，使用逻辑探针接触某一点就能确定该点的状态和能存入脉冲的这两种功能，使逻辑探针成为广泛使用的诊断工具。除了十分复杂的数字电路以外，都能很方便地用它来寻查故障。逻辑探针还能直接显示接触点附近的工作状态，而其他探测仪器通常要借助别的显示设备来观察测试情况。

逻辑探针提供路种指示：

灯光熄灭表示逻辑低电平(逻辑 0)；

灯光明亮表示逻辑高电平(逻辑 1)；

灯光暗淡表示不定或三态；

灯光闪烁表示脉冲信号；

逻辑探测器的电源从测试线路的某电压点通过接线夹子输入，另一个夹子与地接触，这样可提高灵敏度并减少燥声。

逻辑探测器探测低频窄脉冲是很理想的，这种脉冲难於在示波器上看到。而用逻辑探针就能迅速地确定门的输出端挂起或闭锁在高电平或低电平状态。

使用逻辑探针作电路分析一个有效方法是从有疑问的电路的中心开始，检查信号的出现情况。（假设有可供使用的线路图），然后移动探针向后或向前到达有故障的输出端，如图 9－1 所示，这样不需很长的时间就可以找到有毛病的芯片，有故障的芯片的输出端电平不会发生变化的。

逻辑探针的唯一限制是它们只能监测一条线路。

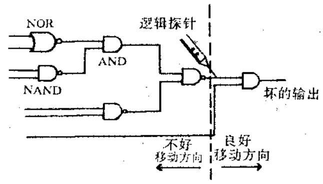


图 9－1 从有怀疑的线路中心开始分析

三．逻辑脉冲发生器

如果被测试的电路没有脉冲或变化的信号，那么可使用一个逻辑脉冲发生器向电路输入可控脉冲。这种轻便的设备就是便携式逻辑脉冲发生器。

当启动按钮开关或滑动开关后，脉冲发生器将识别其探针所接触点的逻辑电平，并自动产生一个或一系列相反的逻辑电平脉冲；该脉冲可以由安装在脉冲发生器手柄上的发光二极管显示出来。

在电路中引进一个变化信号而不需焊脱或切断线路，使逻辑脉冲发生器与逻辑探针成为理想的配套工具。二者同时使用，通过激励/响应来逐段测定某一电路。

图 9－2 表示使用逻辑探针与脉冲发生器测试逻辑门的几种方法。假设与非门的输出端保持高电平，测试输入端 1、2 和 3，发现它们全为高电平，在此种情况下，与门输出端应为高电平，而与非门产生一个低电平输出。如果有毛病的，将逻辑探针放在与门的输出端，发现它为低电平，它本应为高电平，那么，哪一个门是坏的呢？

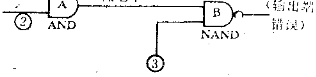


图 9－2 有几种方法测试逻辑门

为了找出毛病，将探针放在与非门（门 B）的输出端，把脉冲发生器置於与门(门 A)输出端(与非门的输入端)，如图 9－3 所示，在该线上加入脉冲。

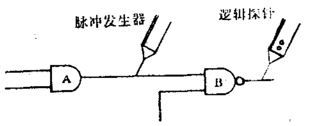


图 9—3 将探针放在与非门输出端上而脉冲发生器放在与门输出端

若逻辑探针闪烁发光，这表示在与非门的输入端有一变化的信号。如果不发出表示变化的光，则与非门可能坏了；但是，如将与门的输出端或与非门的输入端对地短路后，与非门输出端为什么会出现低电平呢？

在与门的输出端放入探针和脉冲发生器，如图 9—4 所示，并注入脉冲信号，如果探针闪烁，则与非门是坏的。因为它的输入端改变状态，它的输出端也应改变状态。

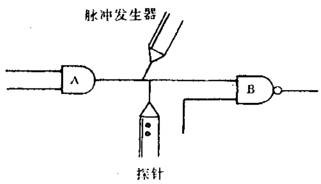


图 9—4 将探针和脉冲发生的放在与门的输出端

如果探针不闪烁，则该电路与地短路。确定哪个芯片短路的方法是用手接触芯片的外壳，短路的芯片是相当热的。虽然某个芯片处在 1 电平的状态似乎是正常的，但正是由于电路某处的对地短路，才使该芯片总处于这种状态而不会改变。

四．电流故障检寻器

第四种轻便型故障检测工具是电流故障检寻探针，这种便携式设备能够确切地判断在苹果机母板上的(或外设板)短路点。电流故障检寻器可以测电路中的电流所产生的磁场。用逻辑脉冲发生器产生脉冲信号，使电流故障检寻器的发光二极管闪烁，表示有电流通过。

如果将检寻器的探针置于印刷电路板的电路上，并且沿着电路导线滑动。只要在所滑动的电路上有电流存在，在探针终端的发光二极管就会闪烁，当滑动通过某个短路点时，则灯光将会暗淡或熄灭，这样就可以找到短路点。

图 9—5 表示在一个逻辑电路中，确定那个门对地短路的一种简易方法：假设 B 门有一个短路输入端，将脉冲发生器和检寻器放於两个门的中间，调整电流故障检寻器的发光二极管刚好发光，向电路注入脉冲，将检寻器的探针置於 A 门的输出端，然后到 B 门的输入端，对地短路的那个门会使检寻器的发光二极管明亮地闪烁，这是因为存在更多的电流由此通地的缘故。在上述检测中，在 B 门的输入端会使检寻器的灯光明亮的闪烁；而在 A 门这一边的线路上不会使检寻器的灯光发亮。根据检寻器上的发光二极管的发光情况，就可以判断电路上电流的情况。

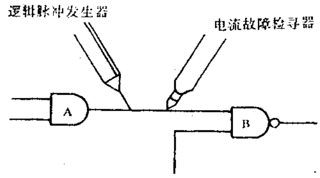


图 9—5 发光二极管发光因短路而发光

五、集成电路测试器

完善的障检测设备是很复杂和昂贵的。一台能够检查苹果机系统上的芯片的检测设备其价格在 1000 美元至 2000 美元之间。价格在 10,000 美元的设备就可以远距离的进行芯片测试处理。

在 Dallas，Texas 的 Micro Sciences，Inc，制造的集成电路测试器可以测试超过 100 种的 7400TTL 和 4000CMOS 系列的器件，这种测试器也能测试部分 RAM 和 ROM 芯片。

在 Gardena，California 的 Microtek Lab 制造的一种测试器可以测试 900 种 54/74TTL 系列芯片的全部功能引线。这种测试器在液晶显示器 (LCD) 上显示被测芯片的情况，而用发光二极管的信号显示测试结果。

San Diego，California 的 VuData Corporation 所推出的这类产品是线路中的元件测试器，它所使用的阴极射线管能显示 50MHZ 的信号，并能显示实际全部电路元件的电压电流特性曲线，这些元件包括电容器、二极管、集成电阻和三极管。使用这种测试器测试的元件情况是由在 CRT 上显示的曲线形状决定的，它能分辨出开路、短路、二极管渗漏、三极管渗漏和隔界的集成电路。这种仪表很有用处，因为它可以测试和挑选安装在电路中的元器件。

六．示波器

如果不是系统设计，而只作一般测试，具备一台 25—30MHZ 的双迹示波器即可满意地进行工作。当然，如果仅为了检测苹果机母板的故障而购置一台示波器，那是很不合算的。

双迹、四迹以至八迹示波器的一个突出的优点是可以同时观察不同信号通道或不同的信号。例如，可以同时观察一个门的输入端和输出端的实际信号，并可以测量一个信号从芯片的输入端到输出端的延迟时间。另一个有效的技术是同时显示数据总线或地址总线的全部或一部分的逻辑电平，从而可以了解在该总线上是一个怎样的二进制数。

七．逻辑分析器

逻辑分析器是与示波器相似的一种设备。它是一台具有存储器的多通道示波器，它收集和存储数字信

号的一个数，并允许同时观察这些信号，如果每个信号是数据总线的一位，那么就可以同时看到整个的数据总线。这就是说我们可以分析任意瞬间在总线上的每一位的逻辑电平，该总线信号被冻结以供显示和分析。这种冻结信号现象或数据模式的性能，对于查寻故障是非常有利的。因为这样就能确定在数字化总线上任何瞬间所出现的信息。

逻辑分析器的工作频率一般从 2MHZ 至 200MHZ。

这种分析器能同时显示许多信号（多输入通道），有一种分析器可处理 32 个通道的输入数据，，工作频率达到 100MHZ。另一种有 48 通道，200MHZ 的分析器。其内部安装了微型计算机和双面双密度软磁盘驱动器。每一个通道都有相应的检测笔夹子连接到线路的每个测试点，检测笔夹子是细小并容易固定的。

目前的逻辑分析器的取样特性已经达到 104 通道，工作频率为 25MHz；另一种为 32 通道 100MHz；还有一种 16 通道 330MHz。另一种结构具有 8 个输入通道而可工作在 600MHz！（请注意 AppleII 的时钟频率为 1.023MHz）。

逻辑分析器在许多场合是很多有效的仪器，例如在调试软件（debugging software）中，读出机器码的数据以及数据在电路的流动情况；可以分析存贮器的输入输出情况，并可同时确定坏的 RAM 芯片；可用分析器来发现间歇性假信号，这些假的尖峰脉冲可能会使系统造成严重的破坏；另外逻辑分析器还可以分析磁盘驱动器的 I/O 操作。

逻辑分析器过去称为数字式示波器，对于软件或硬件设计者来说，它可能是一种很有价值的仪器，但是对于一般的计算机用户，为了维修自已使用的计算机而购置逻辑分析器即不合算也非必要。

八、信号差分析器

逻辑探针可以有效地确定某一个点的逻辑电平和脉冲；示波器扩充了被监测点的数目，甚至可以观测完整的数据脉冲；逻辑分析器更进一步扩展了测试点的数目，它包括了各种大小的数据总线和地址总线。然后，由于这类设备采用了先进技术以及性能的提高，操作这些设备要求具备专门的知识和技术，特别是逻辑分析器，虽其功能很强，但掌握它的操作就更为困难了。信号差分析器的发展就是针对上述情况，它便于用来检测硬件的故障。

信号差分析是一种寻查故障的比较方法，它的工作原理是在被测试的系统中运行一个诊断程序，并且在线路中规定的测试点计算某种编码信号，如果编码信号与从正常系统运行时所获得的编码一致，则在该被测试电路部分未发生故障，当某测试点的信号差与基准正确的编码不一致时，这就表示电路部分是出现故障的区域，然后从该点向前或向后逐一的分析，就可查出出现故障的元器件。

第一种编码是由 Hewlett—Packard 开发的，稍经修改后一直使用至今，这种测试技术的关键是符号编码，它是一种 16 位或 24 位可重复的数位，它表示一种一定时间间隔通过测试点的数据流，这种已知的数据流由一台信号差分析器在一块良好的电路板上的不同位置取样，就形成了在每个测试点的独特的 16 位或 24 位编码。

这种编码以资料方式提供或者存入 PROM 中，在查寻故障中可以反复调用，这个 PROM 成了一个包含正常工作系统的每个取样信号的常规存贮器模块，该正常工作系统是由已知数据流激发或触发的。

信号分析法尚未成为一种通用的查寻故障方法的原因是：它要花费很多的时间去识别测试点或节点，测试这些节点、生成信号、然后记录这种信号编码。当这种工作完成之后，判断故障部位的工作就较为简便了，而采用 PROM 指令模块的方式就可使以上的工作简化。在不久的将来，这种分析技术将会得到进一步的提高与发展。

有一种称为“backtrace”（反向跟踪）方式的分析器，现已有商品生产，它可通过一系列的测试点迅速地查寻故障，指导测试者反向跟踪错误的信号，以寻找故障部分。

信号差分析器的使用是简单的，非技术性的处理就可以查寻故障，甚至未经训练的人员也能使用这种设备和技术。

第二节 元件及其损坏原因

许多故障不要使用昂贵的设备就可找出。事实上，对电子元件损坏的情况有一定的了解就可以修理和排除故障。除操作者的错误之外，通常引起线路故障的原因是使用过度和元件的老化。这些元件包括 RAM 和 ROM 芯片，6502CPU 以及母板和磁盘驱动器之间的 I/O 芯片。CPU 是一种高可靠性器件，一般不会损坏。最易损坏的是其他芯片，除了 ROM 芯片——它们是由苹果计算机公司编程的芯片之外。其他的芯片都是标准的通用器件，它们通常是插接在母板的插座上，其价格便宜且易于更换，另外还有一些焊接在苹果机母板上的元件，它们的更换就较为困难了。

在了解各种维修工具的基础上，进一步分析各类元素的损坏情况，有助于排除故障和修理。

一、集成电路芯片

芯片或集成电路是由硅和其他微量的嵌入硅片某个一定部位的金属（杂质）构成的。由于金属所处的某种位置可以组成微型的三极管，当在芯片的某个地方

加上一个电压时，它可成为一个电压反相器件（+5V，逻辑1可反相为，OV，逻辑0），从而可形成各类逻辑门的功能（与、与非、或、或非门等的功能）。在这种硅片上可以集成极其微小的硅/金属结，已经在很小的芯片上就可集成成千上万只三极管，例如在一块指甲大小的存贮器芯片上就有多达470,000个三极管。

芯片制造厂家的问题是如何将电压和信号输入和输出这样微小的芯片，采用极其细小的导线作为芯片的输入与输出端。导线是粘结或焊接在芯片的微小底座上。导线的另一端焊接支架材料的较大的基座上，支架结构还包括芯片的管脚，用它们可以插入印刷电路板上的芯片插座。

芯片实际上是在有相当大的应力的条件下工作。打开计算机电源时，芯片的温度升高；然后，关闭机器时，芯片的温度下降；再开机器时，温度再次升高，这种热－冷－热的效应，形成所谓"热应力"。它会影响微细的导线束，还包括那些插座的管脚。经过一段时期之后，这种热应力会使导线与芯片和管脚的焊点断裂，这就形成了芯片输入或输出端的开路，这些芯片就要进行更换。

芯片的另一种损坏原因是由所谓"金属漂移"现象引起的。芯片可以认为是一个原子的海洋。某些金属微粒漂浮在这个海洋中，漂移方向与通过芯片的电流相垂直，当这种金属微粒开始在芯片中的某一处集聚时就会发生问题。如果它们集聚在这些微三极管中的某个中间时，就会使这个三极管的工作状态与其他的不同，或不能工作。如果这种集聚的金属具有足够高的电阻，就会使器件间歇性的工作或完全不工作。因为三极管是逻辑门的一个部分，这个门就会出现错误动作，其输出可能是"1"也可能是"0"，不论其输入信号如何。从理论上说，这种消耗损坏需使用几百年后才会发生，但是如果把芯片置于高温、高电压或电源经常打开关闭的环境下，它们的寿命就会缩短，甚至很快就会损坏。

另一些问题出现在芯片的外部，即芯片的导线和被称该器件的输入或输出端的管脚支撑结构之间，这类故障包括：输入端或输出端对地短路，芯片管脚与+5V电源短路，管脚之间短路，管脚开路以及连接点的间歇接触的毛病。最常见的问题是对地开路或短路。在正常使用情况下，芯片最后的损坏是由于某个输入或输出端对地短路。

二、电容器

了解一些有关标准电容器的知识，将有助于我们对这类器件损坏情况的了解。

在第一章中，已介绍在苹果机母板上的几种型式的电容器。一个电容器由两个隔离的金属极板构成，当电压加在这两片极板之间时，电流就立即穿过这个间隙，但是一当电荷层在电极之间形成后，就使电流停止。电容器可以认为是被充电到某个电势。电容器被用来贮蓄电荷和滤除不希望的信号尖峰（突然的、急速的电压尖峰）。

电解电容器是由两片铅箔或铅极板之间隔以浸渍了电解液的多孔性纸构成，在一个电极上（正电极）沉积了一层氧化铝，这就是所谓的电介质，电容器有一个阳极（正电极）和一个阴极（电解液）。电子在一个电极上积累，使它呈负电位，从而将一步阻止电流的通过。

另一种类型的电容器称为"薄膜"电容器，它是在两铅条箔之间隔以塑料绝缘体（通常是聚苯Z烯薄膜）构成，金属箔片作为电极而塑料绝缘体作为电介质，薄膜电容器用环氧树脂封装并有两根铜镀锡的电极引线。

电容器的开路或短路的损坏取决于它们的工作条件和使用期限。

电解电容器特别对老化过程敏感，老化的影响使电解液绝缘体逐渐干燥，开始电容量增加，电气性能下降，最后电容量显著地下降，极板之间相互靠近，引起短路。

另一种损坏的原因是当一些氧化物电介质溶解在电解液中的时候，使电介质层的厚度减少。这种变化通常是由于电解电容器长时期放置不加电压引起的。在这种情况下，电容量增加，电极间形成大量的漏电流，使电容器无法使用。

电容器的引线也可能由於物理性质的作用，使它们与电容极板开路。

由于所加电压过高，使某个电极上的氧化电介质层大面积的剥落，这样也会引起电极间的短路。

同电阻一样，在苹果机的数字逻辑电路中电容器是不大会损坏的。它在电源部分损坏的可能性比在苹果机母板上的较大。但是，过高的温度使电容器损坏的可能性比其他元件更大。

三．电阻

这种限制电流、降低电压的器件是十分可靠的，它将在计算机的寿命期内正常地发挥其功能。但是由于某些因素缩短了芯片的有效寿命，同时也会降低电阻的寿命。高温、高压以及电源的通断对构成电阻的材料有影响。这些应力会引起碳质电阻胶的断裂。或者使碳膜与引线发生开路。特别是由于高压所引起的大电流，会烧毁电阻，但是这是非常少有的情况，因为在数字电路中最高电压是12V（通常为5V）而电流是微安级。

电阻的损坏往往是伴随电路某些其他元件损坏而产生的，打印机电路中的电阻损坏的可能性比主机中

的较大。

四. 电感

电感用於谐电路，其作用是保持电流在一恒定状态。

如前所述，在苹果机母板的色彩调谐电路中有一个小的电感，这种器件是由绝缘导线绕制的线圈。电感的损坏是极为少见的；但是电感如受高温或高压的影响，也可能出现线圈间的短路或线圈的断裂。

五. 二极管和三极管

在苹果机母板上的二极管和三极管均系固体材料制成，其作用是相似的。事实上，三极管可以认为是两个二极管的一部分组成的。

二极管是电路中的一个单向阀，它允许电流按一个方向通过。二极管是由硅或锗制成，它们在电源电路中作整流器，而在某些电路中使电压保持在一恒定水平。另一种二极管由砷化镓制成，当它被某种适当的电压偏置时，可以发光，因而被称为发光二极管或 *LEDs* 。

在计算机电路中，三极管使用在各个位置作为放大器或电子开关。

二极管和三极管的损坏情况和损坏原因与芯片相同。但是芯片的损坏比二极管或三极管更为常见。这是因为在与二极管或三极管大小相同的芯片上，集成了更多的微型三极管，因此，在芯片上会产生更多的热量，其温度会更高。

第三节　使用仪表寻找元件故障

在苹果机母板上的大多数芯片是 TTL（三极管－三极管－逻辑）。如果已知芯片的逻辑功能（与非、或非、或、与等等），当在其输入端加上一个已知逻辑电平的信号就可在其输出端监测其输出信号，从而检测其开路或短路。例如，在两个输入端短接的与门上加一个由＋5V 变至 OV 的慢变化脉冲时，就可在其输出端测到一个与输入脉冲对应的输出电平的变化（脉冲），即当输入端是逻辑“1”时，输出端也成为逻辑“1”(电压在＋5V 和＋2. 4V 之间)。

使用在输入端的工具是逻辑脉冲发生器，在输出端的监测工具则是逻辑探测器。脉冲发生器在芯片的输入端注入一个循环的逻辑电平，而探测器在其输出端测量逻辑信号的有无。

如果该与门的输入端对地短路，则脉冲发生器不能注入信号；从而其输入端保持在逻辑 0 或低电平(接近 OV)。若只有一个输入端接地，也不会使输出端的电平改变而使它保持在逻辑 0。

在门的整个工作时期将门的一个输入端与电源电压短接会使该输入端受到限制或启动的效果，这就是说，即使一种输入状态正确的，另一个输入端每接受一个逻辑“1”，若该输入信号是正确的，则使输出端变为逻辑“1”，这类问题在存贮器电路中已经指出。只有一种输入状的门是一种被短路或开路的特殊门，只要使用这种门就可能产生输出正确或不正确的结果，这个困难的问题将分析如下。

将一个输入管脚短接于＋5V 可能会有潜在的严重结果。当前面的门试图传送一个逻辑“0”或低电平时，这会引起巨大的电流，往往会使该驱动芯片造成灾难性的损坏。当这个输入端对地短路时，也会产生相同的结果，当前面的门试图传送一个逻辑“1”或高电平时，＋5V 的逻辑高电平直接对地短路，形成很大的电流，从而也会产生同样严重的结果。

连接点的开路防碍了逻辑电平的传送和防碍了门对逻辑电平的响应。如果一个两输入端的与非门其中一个输入端开路，如图 9－6 所示，在四种可能的输入组合中，其输出除一种外其他都是正确的。这就是说：当一半的输入端是完好的时候，在大多数时间系统可以正确地工作。但这种类型的故障，会使系统出现间歇性的毛病。

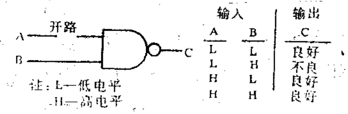


图 9－6　与非门的一个输入端开路，在四种逻辑状态中有一种会出现问题

如图 9－7 所示，如果被测试的器件是一个或非逻辑门，只有当其两个输入端均为逻辑“0”或低电平时，其输出端才会是逻辑“1”或高电平，如果输入端之一开路，它将浮动至逻辑“1”，这样就不会有任何二种输入状态能使其输出端成为逻辑“1”或高电平。因此，其输出将永远处于低电平，正像输出端对地短路一样。

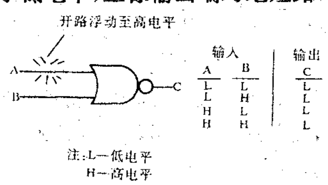


图 9－7　一个输入端开路的或非门，不论输入端状态如何，其输出端总是低电平

如果芯片的输出端管脚断路，它就不可能向下一级门传送任何逻辑“1”或逻辑“0”的电平。测量下一级门的输入端的电压，这时会出现一个逻辑“1”或“高”电平电势（在＋4V 左右）。这里的关键是：在任何时间某个 TTL 门的输入端断路（这种情况称为“浮动”），这个门在它的输入端就恒定地作用了一个逻辑“1”的电平，在这个浮动输入端上的电压将在＋5V 的电源电压和某个电平之间漂移，所谓某个电平是指有效“高”与有效“低”之间的一个电压值，一般约为 1.5V（通常有效

"高"是+2.4V以上，有效"低"是低于+0.4V。）

如果在一个芯片中的某个门的输出端用电压表测量其读数约为+1.7V，这就提示了该输出端是浮动开路，而该电压是由后一级芯片或下一个门所提供的。所有这些故障可以使用一个逻辑脉冲发生器和一个逻辑探测器来检测，并用真空管电压表所测量的电压为辅助手段。

因为苹果机的母板是固定在几个点上的，如果用户在更换芯片在母板的下面没有支撑而下压该板时，可能使电路板上的印刷电路线路断裂和开路。这种细裂缝往往是难於发现的。使用一个放大镜并在强光下观察这种印刷电路板，有时可以发现这种故障。可以用一块三用表或真空管电压表测量电路板线路两端的电阻值，观测表的读数是否是0欧姆；另一种方法是比较线路两端的逻辑状态来查明线路是否断裂。

在测试苹果机母板电路中门的短路或开路时，应特别注意的是一个门可能使用相同的几根输入线由其他的输入或用相同的几根输出线，输出至其他的门，这就是所谓的"扇入"或"扇出"。当研究门电路时，不要忘记：故障可能出现在电路板的另一个侧面。对於一根离芯片很长的线路，它的短路或开路很可能出现在该侧面。这里使用AppleII参考手册中的图纸是很重要的。

（未完待续）

苹果机1.2兆字节控制器通过设计定型鉴定

为满足苹果机用户扩充容量的需要，广东省电子技术研究所研制的ZPCK—6软盘控制器通过了设计定型鉴定。这种控制器与该所ZPC—10A 5吋1.2M软盘驱动器配合使用，能使苹果机由原来的5吋单面单密度每盘容量140K字节，扩充到1.2M字节，填补了我国计算机行业在这方面的空白，为数据量大的苹果机用户带来了福音。

ZPCK—6软磁盘控制器，能控制1～4个ZPC—10A驱动器，能选择单面/双面、单密度/双密度，又能控制8吋与5吋驱动器工作，可用于DOS3.3和CP/M2.2操作系统。

目前，ZPCK—6控制器和ZPC—10A1.2M字节5吋半高驱动器均已批量投产，配成子系统投放市场。

（邱新之）

（上接42页）

周期）。当达到计算值（装入计算值）时，中断标志被置为"1"。此后，内部时钟开始递减计数。直到−255T的最大值。因此在中断标志被设置以后，读操作将告诉你从设置标志到最大值255T间具有多长时间。

系统数据总线用来把数据送到定时器或接受从定时器送出的数据。

寻址方式：

当 $CS_1=1, \overline{CS_2}=\overline{RS}=0$ 时，寻址到RAM。$\overline{CS_1}=\overline{RS}=1, \overline{CS_2}=0$ 时，寻址到I/O和间隔定时器。访问芯片时，$\overline{CS_1}=1, \overline{CS_2}=0$，用 $\overline{RS}$ 脚来识别RAM或是定时器。当 $\overline{RS}=0$ 时，选中RAM；当 $RS=1$ 时选中间隔定时器。用地址线 A_2 来识别定时器和I/O部份，$A_2=1$，访问定时器，$A_2=0$，访问I/O部份。如上表所示：

注：横线"—"为不确定

(a)：$A_3=0$ 表示定时器到 $\overline{IRQ}$ 的中断无效

$A_3=1$ 表示从定时器到 $\overline{IRQ}$ 的中断允许

(b)：$A_1=0$ 表示从 PA_7 到 $\overline{IRQ}$ 的中断无效

$A_1=1$ 表示从 PA_7 到 $\overline{IRQ}$ 的中断允许

(c)：$A_0=0$ 表示选择负沿触发

$A_1=1$ 表示选择正沿触发

果粉藏書

《苹果园》杂志一九八九年征订启事

《苹果园》杂志由中国 Apple 微机协会主办，是为 Apple 微机用户服务的专刊。本刊的宗旨是普及微机知识，推广微机应用，作好技术服务与交流。《苹果园》主要栏目有：国内新闻、技术专文、应用篇、国外动态、使用与维修、娃娃园地、问与答、读者信息等。主要刊登国内外有关 Apple 机的硬件系统、应用程序、监控程序、操作系统、用户要求、使用维修和市场动态等方面文章，另外，从八九年起开辟 IBM－PC 系列机栏目，并承办广告业务。读者对象是从事 Apple 机设计和应用的工程技术人员，科研人员，大专院校和中小学师生，以及有关的管理人员等。

为扩大服务面，欢迎有关单位和个人办理代销业务，单位代销 150 套以上或个人代销 30 套以上者，回扣 20％给代销者。

《苹果园》杂志一九八九年为双月刊，双月出版，全年共六期，每套 5.40 元，单位订阅请填好订单及地址卡从银行汇至山东潍坊市《苹果园》杂志编辑部（开户银行及帐号：市工商行东关办事处 896315）。个人订阅者可从邮局汇款至山东潍坊市东风大街 60 号《苹果园》邮购组，地址卡随信寄来，务请不要电汇或平信夹寄现金。为确保投递准确无误，请务必把地址填写清楚。

中国 APPLE 微机协会《苹果园》杂志编辑部

《苹果园》杂志 1989 年征订单

（代收据）

订阅单位			日期	
地　址			经手人	
发方开户银行帐号	潍坊工商行东关办事处 896315		订户自留	
《苹果园》	每套 5.40 元共(　　)套	金额		
金额(大写)				

此联与银行汇单或邮局汇款收据一并作为报销凭证

果粉藏書

《苹果园》杂志 1989 年征订单

户号：　　　　　　　　　　　　　　　　　　　　（发行存根）

订阅单位				日期	
				经办人	
订户开户银行账号				订户盖章	
《苹果园》	（　　）套	金　额			
金额（大写）					

此联银行位单（或邮局汇款单）一并寄回

户号：　　　　　　　　　　印刷品
通讯地址________________
订阅单位________________
收 件 人________________收
《苹果园》编辑部
第六期
共（　　）份　　潍坊市东风大街 60 号

户号：　　　　　　　　　　印刷品
通讯地址________________
订阅单位________________
收 件 人________________收
《苹果园》编辑部
第 五 期
共（　　）份　　潍坊东风大街 60 号

户号：　　　　　　　　　　印刷品
通讯地址________________
订阅单位________________
收 件 人________________收
《苹果园》编辑部
第 四 期
共（　　）份　　潍坊东风大街 60 号

户号：　　　　　　　　　　印刷品
通讯地址________________
订阅单位________________
收 件 人________________收
《苹果园》编辑部
第 三 期
共（　　）份　　潍坊市东风大街 60 号

户号：　　　　　　　　　　印刷品
通讯地址________________
订阅单位________________
收 件 人________________收
《苹果园》编辑部
第 二 期
共（　　）份　　潍坊市东风大街 60 号

户号：　　　　　　　　　　印刷品
通讯地址________________
订阅单位________________
收 件 人________________收
《苹果园》编辑部
第 一 期
共（　　）份　　潍坊市东风大街 60 号

征订、征稿启事

《苹果园》杂志是Apple微机协会主办，为Apple微机用户服务的专刊。本刊的宗旨是普及微机知识，推广微机应用，作好技术服务与交流、使用维修、信息反馈。并承接广告业务。本刊为双月刊；16开本、每期定价0.90元（含邮费）。全年6期共5.40元。八八年合订本每套5.40元。欢迎广大读者踊跃订阅和投稿。

一、订阅办法

单位订阅请填好订单及地址卡从银行汇至山东潍坊市《苹果园》杂志编辑部(开户银行及帐号：市工商行东关办事处896315)。个人订阅者从邮局汇款至山东潍坊市东风大街60号《苹果园》邮购组。地址卡填写清楚随信寄来，务请不要电汇或平信夹寄现金。

本刊八九年进一步扩大发行、增加印刷份数、欢迎有关单位和个人办理代销手续，单位代销150套以上或个人代30套以上者，回扣20%给代销者，欢迎有关单位及个人承办代销及汇总订阅业务。

二、征稿内容

《苹果园》杂志主要栏目有：国内新闻、技术专文、经验交流、辅助教学、中学生园地、使用与维修、实用程序等。另外，从八九年起，开辟IBM—PC系列微型机栏目。主要刊登对硬、软件分析研究和二次开发的技术专文、各种实用的微机程序、使用维修、编程技巧、程序的局部改进和优化、教学和管理软件，用户处理问题的经验等文章。

三、投稿要求

1. 稿件讲述深入浅出、文字通俗易懂、读者喜闻乐见、易于推广应用。
2. 稿件一定要用20字一行的方格纸书写，每字一格(包括标点符号)、字迹工整、标点正确。不用打印稿或铅印。
3. 按图要在坐标纸上描绘清楚正确，程序要用打印机打印的原件。
4. 翻译或编译稿请附原文(复印件)，并注明文章的出处和作者。
5. 来稿一经采用发表，即付优惠稿酬。切勿一稿两投。作、译者请写明工作单位(全称)及详细通信地址，以使经常联系。来稿若不采用，一般不予退回。
6. 来稿请寄：山东省潍坊市东风大街60号《苹果园》编辑部。

《苹果园》杂志一九八九年征订启事

《苹果园》杂志由中国 Apple 微机协会主办，是为 Apple 微机用户服务的专刊。本刊的宗旨是普及微机知识，推广微机应用，作好技术服务与交流。《苹果园》主要栏目有：国内新闻、技术专文、应用篇、国外动态、使用与维修、娃娃园地、问与答、读者信息等。主要刊登国内外有关 Apple 机的硬件系统、应用程序、监控程序、操作系统、用户要求、使用维修和市场动态等方面文章，另外，从八九年起开辟 IBM－PC 系列机栏目，并承办广告业务。读者对象是从事 Apple 机设计和应用的工程技术人员，科研人员，大专院校和中小学师生，以及有关的管理人员等。

为扩大服务面，欢迎有关单位和个人办理代销业务，单位代销 150 套以上或个人代销 30 套以上者，回扣 20％给代销者。

《苹果园》杂志一九八九年为双月刊，双月出版，全年共六期，每套 5.40 元，单位订阅请填好订单及地址卡从银行汇至山东潍坊市《苹果园》杂志编辑部(开户银行及帐号：市工商行东关办事处 896315)。个人订阅者可从邮局汇款至山东潍坊市东风大街 60 号《苹果园》邮购组，地址卡随信寄来，务请不要电汇或平信夹寄现金。为确保投递准确无误，请务必把地址填写清楚。

中国 APPLE 微机协会《苹果园》杂志编辑部

《结构化 BASIC 语言程序设计基础》出版

该书系统的介绍了结构化 BASIC 程序设计。全书共分七章。第一章讨论了结构化 BASIC 程序的概念；第二章除 CASE 条件结构外，都是原 Applesoft BASIC 语言所提供之指令的介绍；第三章到第七章介绍扩充后的结构化支撑设设施(指令)的内容。其中，第三章讨论了函数和子程序概念。主要是两种子程体的构成和调用。第四章子程序库。介绍了标准子程序库和专用子程序库，子程序库的建立、更新和使用。几种语言的子程序库设施比较。第五章讨论程序的分节与覆盖，介绍了线性结构和覆盖结构。第六章介绍程序的模块化设计方法。第七章集中介绍了支持子程序库设施的 DOS 3.3 扩充命令，是本书的重点。书末附有四个附录。内容包括：结构化 BASIC 版本使用说明，出错信息，程序调试技术简介，常用算法标准子程序示例，以帮助读者建立子程序库的概念，更好的使用结构化 BASIC 语言。该书是在 Apple 机上使用结构化 BASIC 语言的一本较好的参考书。随书同时出售结构化 BASIC 系统软件。该书现已由中国计算机用户协会 Apple 微机协会《苹果园》编辑部出版。欲订购者可与本部联系。